

the world of **68' micros**

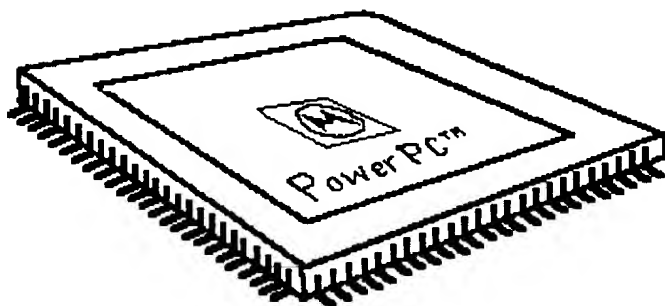
Tandy Color Computer, OS-9, OSK

15 September 1993

Vol. 1 Number 2

\$4.50 Canada, \$4.00 US

The IBM/Motorola/Apple
MPC601 Family of RISC
Microprocessors...



PowerPC™ ...

Perfect for OS-9?

Also in this issue...

Start telecommunicating!
Using internal modems with the PT68K.
We finish the CoCo3Go transportable.

CONTENTS

<i>The Editor Speaks</i>	2
<i>Letters to the Editor</i>	3
<i>Telecommunicating</i>	4
(article) F.G. Swygert	
<i>PT68K Internal Modems</i>	5
(article) Ed Gresick	
<i>Bob Puppo's Amazing KB</i>	5
<i>Adapter</i> (article) F.G. Swygert	
<i>The MPC601</i>	7
(article) F.G. Swygert	
<i>BitBang</i>	8
(article) Richard Kotke	
<i>Video Tape Organizer</i>	10
(program) Jason Reighard	
<i>The Hardware Hacker</i>	10
(column) Dr. Marty Goodman	
<i>Connecting a Modem to CoCo</i>	14
(hardware) F.G. Swygert	
<i>Beginning With OS-9</i>	15
(column) Rick Ulland	
<i>OS-9 Users Group- Alive!</i>	16
(article) Carl Boll	
<i>OS-9/OSK Answers!</i>	17
(column) Joel Hegberg	
<i>Programming in "C"</i>	19
(column) P.J. Ponzo	
<i>Repackaging the CoCo</i>	21
(hardware) F.G. Swygert	
<i>Beginner's Showcase</i>	23
<i>Telecom</i>	24
<i>Micro News</i>	25
<i>Swap Shop Classifieds</i>	25
<i>Reviews</i>	26
<i>SuperComm</i>	
<i>OS-9 Disk Mailing List</i>	
<i>This Issue's microdisk</i>	26
<i>Submitting Material</i>	28
<i>Advertiser's Index</i>	29

POSTMASTER:

If undeliverable return to:
FARNA Systems PB
Box 321
Warner Robins, GA 31099-032

the world of 68' micros

Published by:
FARNA Systems
P.O. Box 321
Warner Robins, GA 31099-0321

Editor: F. G. Swygert

Subscriptions:
\$23/year (8 issues)
\$30/year in Canada/Mexico
(\$12 US, \$16 CIM for four issues)
overseas subscriptions \$35/year
(\$18 for four issues)

Advertising Rates:

Personal Classifieds- \$0.20/word, \$4 minimum. Name, address, and phone number free. FREE for subscribers.

Display Ads- \$15 1/6 page, \$20 1/4 page, \$35 1/2 page, \$60 full page, copy ready. Add \$10 for special placement, \$10 for typesetting (\$5 1/4 or less). Dot matrix will be typeset if deemed unacceptable and submitter billed. 10% discount for four or more appearances.

All trademarks/names property of their respective owners.

The publisher welcomes any and all contributions. Published material becomes the property of FARNA Systems unless other, prior arrangements made. Submission constitutes warranty on part of the author that the work is original and not copywritten by another party. All opinions expressed herein are those of the individual writers, not necessarily the publisher or editor. FARNA Systems reserves the right to edit or reject any submitted material without explanation. Renumeration discussed on an individual basis.

Back issues are \$3.95 plus \$1 shipping for one issue/copy. Add \$0.50 each additional.

Newsstand/bulk orders available. Dealers should contact the publisher for details.

Problems with delivery, change of address, subscriptions, or advertisers should be sent to the publisher with a short description.

The publisher is available for comment via Internet (dsrtfox@Delphi.com) or Delphi E-mail (DSRTFOX). The CoCo and OS-9 SIGs on Delphi are also frequented (The Delphi SIGs are sponsored by Falsoft).

ENTIRE CONTENTS COPYRIGHT
1993, FARNA Systems

The editor speaks...

F.G. Swygert

I just finished mailing over 100 copies of the first issue. In all, 150 copies were printed (first issues were sent to all who subscribed before 30 July, and 12 were sent to writers and advertisers). I am very pleased with this, as most subscribed in blind faith that a publication would actually be put out. If reaction is anything like that of my colleagues in my "normal job", I expect to double that number over the next four months, possibly triple by the beginning of the new year.

All this is possible only with your continued support. Don't forget to drop me a note and let me know what you think! I take all criticism constructively. If you want a personal reply, include a self addressed, stamped envelope. I'll take the time to hand write a short note, and possibly print your letter too. Due to space restrictions, all letters can't be printed.

Just in case any of you are interested, my "normal job" is with the US Air Force. I am an active duty Staff Sergeant stationed at Robins AFB, Georgia. This is about 15 miles south of Macon, 100 miles south of Atlanta. My current duty is an assistant system operator of the Civil Engineering department computer system- a Wang VS 8400. This is a large (approximately 4'x4'x3') mini computer with over 200 terminals and 50 other devices connected to it. Some terminal cables run nearly 1/3 mile! The operating system is, in many ways, similar to OS-9, at least in principal. With this monster, you never really see the operating system. All functions are handled through menu driven utilities!

My main computer at home is a CoCo 3 with two 40 track double sided disk drives, Commodore 1084S RGB monitor (a badge engineered Magnavox 8CM515), 512K, Maxon 2400 modem, and various other items. You will see this computer in upcoming issues as I repackaged it, a Disto SCII, and SCSI hard drive into a Tandy 2000 case. It is currently packaged in a home made case. The portable on the cover of last issue is mine also, so I am very experienced in repackaging. Note that I took the easy way out each time though... no MPI!

My secondary computer, the one this magazine and many books have been produced on, is an IBM clone with an 80286 20MHz processor. The only purpose for this machine is to run Aldus PageMaker and a few associated utilities. PageMaker was expensive, but I don't own a laser printer. I take a disk copy of the magazine to my printer, who uses a Macintosh with the same version of PageMaker (4.0). When I can justify the expense of my own laser and a new computer, I will look into buying a 68000 based

machine, if there is a decent desk top publishing package out in the next couple years.

Telecommunications, the theme for this issue, is vital to the continued support of the Color Computer, OS-9, and even the newer 68000 based machines. Through a modem, support can be found just moments away. Don't think you can afford a major system like Delphi or GENie (I won't mention CompuServe... they *are* to expensive compared to the others!)? Think again! If you are spending long distance phone calls to connect with a "local" board, you would probably save by subscribing to one of these systems. The new rates are as low as \$1 per hour.

If there isn't an access number to either, then try to find a FIDO board that carries the CoCo or OS-9 echoes. If there is a few people in your area interested, any FIDO network board can carry the echo for you. Just remember that the system operator will expect the echo to generate some traffic or he will be forced to drop it. Carrying the echo does mean a bit of work for the sysop. Remember too, that if you subscribe to a long distance company such as Sprint or AT&T's *Reach Out America* plan, it is cheaper to call a neighboring state than to call long distance to the next county!

Speaking of "reaching out", I just received word that a bi-lateral printing agreement I proposed with "CoCo-Link", an Australian CoCo magazine, has been accepted. Basically, Fred Remin (CoCo-Link editor/publisher) and I have agreed to allow reprinting of articles from the each other's magazines. This benefits the publishers by having a ready source of fresh material. The subscribers benefit from not having to pay expensive airmail delivery rates and from getting news, programs, and articles they may never otherwise hear of. In all, I think everyone will be pleased with the results.

Most of you should remember my "pet" abbreviation for the magazine from the last issue (268'm). Well, you will see it often from now on! It was suggested that I use an "end of article" marker of some sort. What better than the abbreviation? For those who didn't get the last issue, the "2" comes from the first letters of "the world of".

< 268'm >

Coming next Issue...

- * OS-9 in Industry
- * PT68K cards that work

VENDORS - Note the new ad rates!
SUBSCRIBERS - classifieds are now free with no limitations!

Letters to the Editor

The following is an edited letter to the editor from Ed Gresick, owner of Delmar Systems, distributor of PT68K (System IV) computers, referring to continued support of OS-9. The original post is on Delphi, number 78853, OS-9 SIG forum:

Let me address the issue of software pricing first. I recognize there are people out there (regardless of hardware or OS) that are on a very tight budget and can't afford several hundred dollars for a software package. But, many more can. Many of those who protest the price of OS-9 software also have IBM compatibles. How much are they paying for software for them? (The argument of piracy is really a non-issue; the individual who justifies stealing a \$500.00 package because of its cost is just as likely to steal a \$30.00 package rationalizing that \$30.00 isn't going to hurt anyone.) Until we recognize that programmers deserve compensation for their efforts, we aren't going to attract the new programming blood we so desperately need to OS-9/OS-9000. I do empathize with those who really can't afford the price of some of the software. There are many things I'd like to buy but I can't afford them either. Let us not restrict our thinking to making price the common denominator. Availability and quality must be paramount.

I use a varying number of experienced OS-9 programmers at times to do special contract jobs for me. Several of them wrote software for the CoCo; some here may be using some of the software. I've talked to them regarding writing software for OS-9(68000). Common to their answers was the fact that, in their experience, the users were not willing to pay enough to make it worth while for them to make the effort.

Now I'll rattle a few cages. Supposing, by some magical means, we suddenly had a wide variety of software comparable to and priced competitively (or even lower) to what's available for MSDOS. What would be the rational for customers to buy an OS-9/OS-9000 platform vs an MSDOS platform? Explain to me, in terms I can use to the average customer coming to my store, what advantage(s) OS-9/OS-9000 has for him vs MSDOS (I'll even allow that the hardware cost is the same). Considering how solidly MSDOS is entrenched, I suspect the case will be very difficult to make. As I said before, that window of opportunity is closing.

OK, slightly different tack - what

can OS-9/OS-9000 do better for the average customer than MSDOS can (even allowing for WINDOWS, OS-2, etc.)? Until we find the answers to that question we won't progress expanding the user base of OS-9/OS-9000 in the home market.

I'll throw a couple of thoughts out. There are many households with 2, 3 or more computers in use. There are even more households where the additional computers are on the shelf collecting dust. In addition to MSDOS machines, these would include CoCos, Commodores, Apple IIs, etc. Let the main computer be an OS-9/OS-9000 machine and hook up the other machines as terminals. Now everyone, Mom, Pop and the kids can access and use the main computer and the software simultaneously. We can add a little plus for Mom (or the family cook). A few years back, Tandy included a nice menuing/recipe program with DESKMATE. This kind of software and a touchscreen in the kitchen makes for a nice plus.

Implicit is the requirement for quality software that has the look and feel of the better MSDOS software. For starters, we'd have to have a WP like Word Perfect or MS Word, a spreadsheet like Lotus 123 or EXCELL and something like QUICKEN. Other packages would have to follow quickly. Pricing could easily be comparable to the equivalent MSDOS product and since it's multi-user, could be higher.

What I'm suggesting is to exploit the multi-user (and to a lesser extent the real-time) capabilities of OS-9/OS-9000. Compared to UNIX, OS-9/OS-9000 is less expensive and much easier to set up and administer. In fact, I think a fairly simple program can be written to handle what little administration and maintenance is required.

The above thoughts aren't the only way to enter the home market nor necessarily the best way. Others may have better ideas. I am sure as to the market segment to address; i.e., either through businesses or the more affluent individuals - but we must have something to offer them beyond what they can get from MSDOS.

If you're going directly after the home market, the initial market thrust would have to be towards the upper middle income levels - say that the market potential is at least 1 million. If 1 percent of these bought the first year, I'd say the effort was very successful. Even 0.2% the first year would be a credible showing.

You, as the publisher of a maga-

zine serving this market (and the other publishers) can influence the thinking and direction we take. I suggest you do your own investigation as to how to reach the home market and what is required. Please do not take my word for it - I've become very opinionated and crotchety about this subject! Then 'editorialize' accordingly. And you can support the few who are trying to spread the word about OS-9/OS-9000 on other forums, BBSees, etc.

Ed Gresick - Delmar Co.
Box 78
Middletown, DE 19709

Ed has given us all a lot to think about! Even though the letter is targeted at OS-9, much of it applies to DECB or any other operating system.

I received the first issue of "68' micros" yesterday and to say I am pleased would be a gross understatement. I particularly like your article on "C" and Rick Ulland's column "Beginning OS-9". Never have I seen OS-9 defined so succinctly and explained so well. Even learned what the BLOB is. In my short career in OS-9, I've seen it referred to many times and have fallen victim to it to often; and now I know (well, sort of) what it is. JOY!

I was also somewhat chagrined to see the ad for your book, "Tandy's Little Wonder", because I had forgotten about it. Anyhow, enclosed is a check for the book. Since I'm so late with this order, I won't insist on an autographed copy!

Mel Machesney
1774 Arlington Blvd.
Huntington, WV 25705-2705

Thanks for the kind comments Mel! And since an autograph seems important, I sent that along also. Did anyone else forget to order? I have several left and can print more!

I just got the first issue of "68' micros" must say that I am quite impressed! It seems to be very well put together, has a lot of the flavor of magazines we just haven't been seeing. It's really a refreshing change to actually see something that has a mixture of BASIC, OS-9, and lots of ads. Just wanted to let you know I got the first issue and it looks real nice!

Allen Huffman - Sub-Etha
Box 152442
Lufkin, TX 75915

Allen also passed along a couple tips on making the magazine look

better. Those tips are incorporated into this issue. If anyone sees something that can be improved, please pass it along! I may not agree and keep things as they are, but then you may catch something I missed!

Seems to me like it would be good for the CoCo (& OS-9/OSK community) if you, JWT, (OS-9 Underground), & "No-Name" could merge into one strong CoCo publication!

Ron McCauley
201 Trealout #17A
Fenton, MI 48430

Ron, you probably aren't the first to think this! It would be nice for the community, but the editors of each publication would have to give up some of their editorial powers. Also, there would be the question of who would control funds and distribute profits, control magazine layout, etc. I'm sure each of the editors in question believes he is the best to handle those duties! Each of us worked hard and would be unwilling to give our products to someone else at this point. Of course, if one of the others disagree, I am more than willing to discuss a potential merger.

What you suggest is simple on the surface, but a nightmare to accomplish. It could be done, and maybe it will one day, but right now everyone is jockeying to establish a position for themselves. Eventually, one or more of us will fall by the wayside and quit, leaving one or more of the others to continue on, just as has happened earlier in the life of the CoCo when Hot CoCo, Color Computer Magazine, Rainbow, etc. were all being produced at the same time.

All I can say at this time is that I committed to produce "68' micros" for an absolute minimum of two years, so there will definitely be 14 issues after this one. After that, publication will continue as long as there are enough subscribers to make it profitable and rewarding - I have no doubts this will be for several more years. I have made plans to secure a future for "68' micros", and I hope you and all the other subscribers will still be with us in the future!

Letters are printed on a space available and popular subject matter basis. If you don't want your letter printed, or wish to withhold your address or name and address, please state so when writing. In some cases, letters are slightly edited for space and/or clarity. If a personal reply is desired, please enclose an SASE.

The following will start with a pretty basic explanation of just what telecommunications mean to computer users. Advanced users may want to skim over the first part, but do pay attention to the file compression section. Some of these utilities will be available on "microdisk". Discussion of file types that can be decompressed by them and the machines they run on are in the later part of this article

The word "telecommunications" actually means communication at a distance (as by cable, radio, telegraph, telephone, or television) [Webster's Seventh New Collegiate Dictionary]. So any time you talk on the phone, you are actually telecommunicating! With the advent of the computer, the word has gained a more common meaning- one computer communicating with another over the telephone lines.

Why would you want to start telecommunicating with your computer? Read the "telecommunications" columns. A LOT of free software and shareware is available on bulletin boards and networks around the country for almost any type of computer. Help can be found within days, if not immediately. Simply leave a message requesting help then check back in a couple days for an answer. Some questions take longer, but most will have several responses over a few days. Much depends on where the message was left. Local boards may have very little traffic for the CoCo and OS-9, but the networks and pay services usually have a good deal. The "Telecommunications" columns will help with more information.

So what is needed to start telecommunicating? Obviously a computer and a telephone, which anyone reading this most likely has already. Since most computers aren't equipped to make sounds that can be easily transmitted over the phone and understood by another computer, a device called a modem is also needed, as well as terminal emulation software.

The word modem is an acronym for "modulator-demodulator". This device acts as a translator between the computer and telephone system. All computers have a basic common "language"- they are all binary devices. They have only two "sounds" to their language- a one (1) meaning "on", and a zero (0) meaning off. All characters, symbols, and com-

mands that any computer understands is made up of combinations of ones and zeros. The modem modulates the ones and zeros (turns them into tones) so they can be easily transmitted over telephone lines. The receiving modem the demodulates the tones back into ones and zeros that the receiving computer can store.

Any computer can transmit or receive the code of any other computer, provided the transmitting computer can read the code from the disk. Therefore, a CoCo can easily receive a file, even a machine language file, meant for an IBM PC clone, MM/1, or any other computer. The file won't be usable unless it is in some type of code the CoCo itself can comprehend (such as ASCII text files) though. One can, however, use the file on the machine it was intended for IF it can be transferred to a disk format the intended machine can read. If you have a utility that allows reading a CoCo disk for your PC, the file can be transferred to the PC and it should run. The CoCo OS-9 PCDOS command will only transfer ASCII files, however.

What Kind of Modem?

There are two basic types of modems- internal and external. Almost all OS-9 computers and the CoCo require an external modem. An internal modem lacks a case and power supply, that is why it is usually cheaper than a comparable external unit. There is no way to use an internal modem with a CoCo or most OS-9 machines. These modems are designed for the ISA standard PC expansion bus.

The exception is the Peripheral Technologies PT68K series (used in the Delmar Co. System IV). These computers were designed with a standard PC bus so they could use inexpensive expansion cards. Most internal modems should work in these computers, but a modification to the IRQ line is required. See sidebar for details.

For the most part, get as fast a modem as possible. 2400 baud is adequate, but downloads will obviously be take less time with a faster modem. Many CoCo users will be pleased with 1200 baud units, which can often be picked up used for under \$25, since CoCo files aren't as large as those for other computers. Downloading ANY file will take a long time using an ancient 300 baud modem! They are adequate for reading mail and messages

until beginners can afford/locate a faster one, but that's about all. Note that most pay services charge extra for access over 2400 baud (less time on-line for you means less money for them!). So read the messages and mail at 2400, then log back on to download files at the faster rates.

See the "Hardware Hacker" column for details on modems and connecting to the CoCo. CoCo OS-9 users may also find the "Bitbang" article of interest.

Terminal Emulation Software

There are many different terminal emulation packages for all computers. This software processes information being sent between computers. They are like word processors in a way... different ones have different features, ranging from basic to elaborate. I can't discuss which packages are best in the space provided. Such a discussion would take an entire issue or two! One will have to try several packages based on advertised features and choose the one they like.

What I have done is included a basic terminal program for CoCo OS-9, OS-9/68000, and DECB on this issue of "microdisk". These programs will get one connected, where other programs can be downloaded and experimented with.

What's A "Download" and "Upload"?

The act of transferring a file from a remote computer to your computer is called "downloading". Transferring from your computer to a remote is "uploading". These terms are often abbreviated as "u/l" and "d/l", with an "d" added for past tense ("u/l'd" or "d/l'd").

There are several protocols used to transfer files. All have some sort of handshaking arrangements between the computers so each computer will know when to send/receive another portion of the file. Choose a protocol that the host computer and your terminal software both support. Generally, the higher the letter that the protocol starts with, the faster it is (Y modem is faster than X modem). I won't go into any details at this time, as there are many different protocols available. Some services, such as Delphi, call Xmodem-1K Ymodem, which can be confusing. Experiment with the different protocols. Maybe a future issue will have an article covering the many protocols and their differences if readers request it.

What is an ".ARC" file?

This term refers to compressed, or "ARChived", files. Different compression utilities have their own file extension for easy identification. These utilities take a file or group of files and compress them into a single smaller file. There are several different compression methods, with each utility using a different method.

The AR utility is available for CoCo OS-9 and OS-9/68000. Each version (CoCo or 68000) will explode files compressed with the other (AR 1.2 and later). PAK is for CoCo OS-9 only.

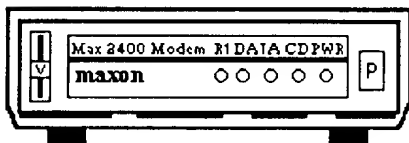
The most common compression utility for Disk BASIC is TC (The Compressor). TC isn't quite as efficient as the OS-9 utilities, but works quite well. There are currently three versions of TC. TC12 is for the CoCo 1 and 2, TC3 for the CoCo 3, and TC31 for a CoCo 3 with 512K or more. Each version should explode the other's files. TC will not compress an entire 68 granule disk. The limit is somewhere around half a disk to be compressed as a single file. TC31 will compress a larger amount of disk space. Versions of AR for the CoCo and OSK and all three versions of TC will be found on this issue of microdisk.

E-Mail and Forums

There are two ways to leave messages on a bulletin board system: E-mail (electronic mail) and the public message base. Some boards simply call the public area the message area while others have special names for it (Forum, Round Table, etc.). The main difference is that E-mail is private to the individual or list of individuals you send a message to. Anyone can read a public area message whether it is addressed to an individual or "all". The system will usually inform an individual when a person has public messages directed to them.

Well, that about wraps up a brief overview of computer telecommunications. If there are any questions, don't forget to write! So what are you waiting for? Get a modem and get on-line!

< 268'm >



PT68K (Delmar) Internal Modems

Delmar Company currently offers only one FAX/MODEM card. It is the model VFP from ZOOM and is modified by Delmar technicians to work with the System IV and V. Price is \$289 which includes the driver and descriptor. Warranty is 7 years and must be handled by Delmar. STerm, ATerm and Kermit all work fine. FAX software is not available yet but is in the works. A FAX viewer for G-Windows should be available for the third-party FAX software. A low cost internal modem hasn't been selected yet, but will probably be a ZOOM since they allow Delmar to keep the seven year warranty after modifications have been made.

Modifying an internal modem for Delmar and PT68K computers is not difficult but does require making 2 solder connections to the modem card. Making these connections will void the warranty. All that's required is a 7406 hex inverter, a 470 ohm resistor and a 2-pin Berg connector. Straighten out all the pins of the 7406 except 7 and 14. Solder pin 14 of the 7406 to +5 volts and pin 7 to ground. This may be done by piggy-backing onto an existing 14 pin IC quite easily. Make sure none of the other pins of the 7406 touch the other IC. Only one of the inverters is used (say pins 5 and 6). The unused pins may be clipped.

Connect the 470 ohm resistor from pin 5 to pin 7 of the 7406. Run a wire from pin 5 of the 7406 to one side of the 2-pin Berg connector and a second wire from pin 6 of the 7406 to the other side of the 2-pin Berg connector. Remove the IRQ selection jumper from the IRQ selection pins on the modem card and replace it with the 2-pin Berg connector. One side of the IRQ selection pins are tied together and go back to the IRQ generating circuits on the board. The pins on the other side are connected directly to the card edge connector. The side of the 2-Berg connector connected to pin 6 of the 7406 IC should go to the side of the IRQ pins going to the card edge connector. A driver and descriptor is available for those who want to do their own modification for \$50.00. Note that any modem purchased from Delmar will have this modification already intact AND be fully warranted. This modification is only guaranteed to work on a ZOOM VFP Fax/Modem, but should work on almost any internal modem.

Ed Gresick
Delmar Company
Box 78
Middletown, DE 19709
(302-378-2555/FAX 2556)

EDITOR'S NOTE: This modification should work on Peripheral Technology PT68K2 boards as well. The Delmar System IV uses a PT68K4 board.

< 268'm >

Bob Puppo's Amazing Keyboard Adapter!

F.G. Swygert

Years ago, a young electronic tinkerer (Bob Puppo) decided his CoCo would be better off with an IBM style keyboard. The problem was that the CoCo used a wired matrix to communicate keyboard input to the CPU, while the IBM/PC used an intelligent keyboard with a simple serial connection. The serial signals delivered from the PC keyboard had to be decoded then sent to the appropriate connections in the CoCo. The simplest method to accomplish this was to use another microprocessor to do the work, since having the CoCo do all the work would slow it down and require extensive ROM changes.

Bob's first effort used an Intel 8051 microcontroller. This unit has the CPU and ROM all built into a single package. The result worked so well that he decided to market his amazing little device.

He first tried to interest Owl-Ware, who sat on it. When that failed, he decided to market it himself.

The first commercial model used the MC68701 microcontroller, similar to the 8051. After only a handful were sold, Bob's source for low cost MCUs dried up. The circuit was redesigned to use a 6802 CPU and separate 4K (2732) ROM. The majority of the units sold used this configuration.

The keyboard interface was rather expensive at \$100+. After a short while, the market simply dried up.

It has been several years since these interfaces were sold, yet there is an occasional quest to find one or one needs repair. Bob has graciously allowed "68' micros" to print the schematic so that readers can repair broken units or build their own. The ROM code is on this issue's "microdisk".

PARTS LIST:

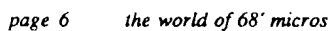
U1 - MC6802	U2 - 2732/64 EPROM
U3 - 74HCT238	U4 - 4094
U5 - 4013 Q1 - 2N3904	
D1 - 1N914	D2 - 1N100A germ.*
C1 - 100uf 10V	C2, C3 - 5-20pf 10V
C5-C8 - .1uf 10V	R1 - 10K
R2 - 1K	XTAL1 - 3.579MHz
U6-U9 - MC142100	
VR1 - 7805T with heat sink	
J1 - Connector for CoCo keyboard	
J2 - 5 pin DIN socket, IBM/PC keyboard	
C4 - .001uf (use only if 4013=squared d)	

*D2 MUST be germanium!

A board could be wire-wrapped, but that would be quite a chore! A source for bare boards and possibly partial kits will be announced in the next issue. Have to give the supplier time to create a board from scratch and get a few produced first! These WILL NOT be copies of the original board. They will most likely be slightly larger.

Mr. Puppo gives permission to use the schematics for repair or personal building ONLY. Producing for resale is strictly forbidden and would result in infringement on Mr. Puppo's design copyright.

< 268'm >



A new processor from Motorola, IBM, and Apple

Motorola, IBM, and Apple teamed up to produce a better microprocessor for their future PCs. These CPUs will have the capability of running UNIX, DOS, Windows, Macintosh System 7, and several other operating systems. With any luck, Microware will add OS-9 to that list, although the current Mac version of OS-9 will run (along with 90% of the current Mac software).

How is this possible? Through RISC (Reduced Instruction Set Chip) technology and emulation. The chips are built using Motorola technology and IBM POWER (Performance Optimized With Enhanced RISC) architecture. The 66MHz MPC601 is powerful enough to emulate a 25MHz MC68040. Those familiar with running the 6809 emulator under a 16MHz 68000 can appreciate this performance. This is similar to the Mac Quadra 700.

The MC601 itself is manufactured by IBM. Future chips will be manufactured by Motorola. It is based on 64 bit architecture, but the 601 is a 32 bit implementation. 2.8 million transistors are packed onto an 11mm square four-layer metal CMOS substrate with a density of 0.65 microns (the smallest distance between traces). By comparison, the Intel Pentium uses three-layer, BiCMOS 0.8 micron technology to put it's 3.1 million transistors into a 16.6mm x 17.6mm package.

The new processor has a 32 bit external address bus capable of accessing 4GB of physical memory, 64 bit external data bus, and internal data bus widths ranging from 32 to 256 bits. The integral MMU (Memory Management Unit) up to 4TB (terabytes) of virtual memory through 52 bit virtual addresses. The MPC601 is also faster, uses less power, and cheaper

than the Pentium. Bench marks indicate that the MPC601 is 1.44 time faster on numeric sorts, up to 4.7 times faster on bit-field operations, and FPU operations were 1.53 times faster. Tests were conducted on a 60MHz MPC601 system and a 66MHz Pentium system. Parallel processing is supported in all versions of the PowerPC chip.

The partnership wanted compatibility with existing software to help the new processor gain immediate market acceptance. Much of this compatibility is accomplished through emulation. It is possible to have UNIX (actually AIX-IBM's current UNIX implementation), Windows, DOS, and Macintosh System 7 all running on the SAME computer at the same time. Simply switch between them much as one would switch between windows.

The 680x0 emulation follows the MC68LC040 processor. FPU (floating point unit... the math coprocessor) and MMU functions are not emulated. Programmers must either follow the Standard Apple Numeric Environment (SANE) which is built into System 7 or use the MPC601 floating point instructions. MPC601 MMU functions must be used.

Of 600 current Mac applications Apple tested, 90% ran flawlessly under the emulator. This high percentage and the speed of the emulator is due to the fact that most Mac applications spend 60-90% of their time in the Mac Toolbox, which is built into System 7. The portions of the Toolbox most often used are written in native MPC601 code, much as a BASIC programmer would use assembly language for routines that needed to run fast. Eventually, all of System 7 will be ported to native code. Native 601 code should run two to four times faster than that written for the 33MHz MC68040 based Quadra 950.

To keep things simple for future programs, the same code can be written to run on a 680x0 or MPC601 based Macintosh. This is done through resource code that has pointers showing the locations of the 680x0 and MPC601 binary code segments. The program must be written in full 680x0 compatible code, then routines in MPC601 code added. The MPC601 version of System 7 will switch between the 601 and 680x0 code segments as needed. The application will be larger, but can be packaged on the

The PowerPC Family

MPC601

32 bit address bus, 64 bit data bus. Processor bus based on Motorola MC88110. Available now.

MPC603

Notebook version of 601. Lower power consumption, fully static logic. Available in mid 1994.

MPC604

Bigger pipeline, higher parallelism, and advanced branching techniques for higher performance. Should be available in mid 1994.

MPC620

Full 64 bit address and data bus versions featuring multiple levels of parallelism. This version will use a different processor bus. This version is targeted at high-end workstations and network servers. Should be available late in 1994.

same disks rather than having two separate versions.

Intel 80x86 and Windows compatibility are handled through emulators as well. A port of Windows to native MPC601 code is in the works though. Since Windows operation is similar to System 7 in that an application spends a great deal of time within the Windows environment, this should be very fast.

The IBM/Apple/Motorola alliance was formed in October 1991. The goals of the alliance are to consolidate research and development on future software technologies, promote open systems, construct new processors, and to cut development costs. An underlying goal is to seize the lead in the desktop PC market.

If the processors are up to what has been promised, I think they are on their way to causing Intel major headaches. And if Intel thinks they have problems now, wait until Microware ports OS-9000 to the PowerPC family... if they ever do! For more information, contact Motorola, IBM, or pick up the August 93 issue of BYTE Magazine. Oh yeah, scream at Microware about a possible port while you're writing for info, okay?

< 268'm >



The bit of silicon the MPC601 is built on is only this big and contains 2.8 million transistors!

Processors Compared

	MPC601	Pentium
Instruction Architecture	RISC	CISC/RISC
Max. instructions per cycle	3	2
Built in cache	32KB unified	8KB instruction 8KB data
User registers	32 GPR 32 FPR	8 GPR FP stack
Ext. address bus	32 bits	
External data bus	64 bits	
Word size	32 bits	
Speed compared	66MHz	
Power use	9W@3.6V	13W@5V
Cost per 1000 (approx.)	\$450	\$965

The History of Bitbang

A long time ago in a galaxy far far away there was a college student who had a computer. Of course, it was a Tandy Color Computer, a really nifty 64K D-board CoCo-I which was black because he removed the ugly grey paint with paint thinner.

This college student was very excited! He had just gotten his first disk drive! The drive cost as much as the whole computer had. He also had a copy of OS-9 level I and was enthusiastically playing with it. "I know it is a multi-user system," he said, "but I can't prove this to anybody unless I can run a terminal off of the serial port. Alas, if only I had enough money for a Multi-Pak and an RS-232 pak; but I am just a poor college student, working part time to feed myself..."

That's how it began. The first version of bitbang was for level I; I had it working reliably at 9600 baud using an old Lear-Sigler ADM 3A "dumb" terminal. It used a jumper in the serial cable to provide a FIRQ interrupt and when the driver got the interrupt it timed the rest of the serial data bits. It was also really only "half duplex" since it wouldn't detect incoming data if it was in the middle of transmitting data.

In 1985 there were no terminal programs for OS-9 so I just toyed with the multi-user facet. I easily impressed a friend who owned an Apple II (remember those?) by hooking it up as a data terminal to my CoCo I.

In 1987 I got a CoCo 3 with 128K. The old Level I bitbang driver wouldn't work. Every time data came in, the screen exploded into a colorful lockup. What was wrong? Well, as it turned out OS-9 Level 2 didn't like the FIRQ very much; it was totally unsupported. I never really got it working. It would receive a few characters and then lock up; I had to give up temporarily and return to my studies.

I graduated in 1988 with a BSEE and an additional major in Computer Science. Since I was on a Naval ROTC scholarship I was commissioned as an Ensign, USN and sent off to nuclear power school in Orlando, FL.

Finally I completed my "nuke" training and was sent to Surface Warfare Officer's School in San Diego, CA. Compared to nuke school this was a six month vacation; I had plenty of time to start chasing the FIRQ bug.

What I found was that sometimes the FIRQ came through the PIA as a FIRQ, and at other times it was mapped by the GIME as an IRQ. Additionally, because page flipping and other things had to be done prior to jumping to the interrupt routine (because the system memory map is different from a given process's) I couldn't "cheat" and jump di-

rectly to the driver. I had to use OS-9's built in interrupt vectoring routines. This would slow the interrupt and add in a large random factor on the response time, so operation above 1200 baud was not feasible.

I got around the FIRQ/IRQ problem by writing a patch to the kernel so that FIRQs are treated as IRQs (by stacking all the registers in the proper order). Since I didn't have I good disassembler at the time, I had the bitbang driver kludge some code onto the end of OS9P1. This caused a weird bug that I didn't notice until a year later: sometimes every sixteenth character read by I\$read would be wrong, but I\$readln would work OK. Since I didn't know how to fix that bug and it took so many man-hours to get the driver working in the first place, I just put a caveat in the docs and left it at that.

The first BBS I called with bitbang was the Ocean Beach BBS in Ocean Beach, CA. I had some trouble downloading so I went back into bitbang. I expanded the receive buffer to 2k and had it wait a few milliseconds after a byte was received to see if more was coming. That fixed the problem, so I uploaded the first version in October, 1989.

The weird bug and the kludgy patch have always bothered me. I was at sea for most (75%) of 1990, 1991, and 1992 onboard the USS TRUXTUN (CGN-35). We were homeported in Bremerton, WA (a 1 hour ferry ride from Seattle). I had my computer set up in my stateroom but because I had no outside phone lines I didn't use bitbang, plus I was nearly as busy as when I was in nuke school. Workweeks run to 100 hours when the ship is underway, plus you stand watch for 5 hours a day on a rotating basis. In port you stand 1 duty day every three and you're on the ship overnight on duty days.

I transferred off of TRUXTUN in February 1993 to my present duty station in Green Bay, WI where I train sailors at a nearby shipyard. This is shore duty so it's pretty laid back - 8 or 9 hours a day, weekends off, and occasionally half of Friday off. So once again I had the time to chase bugs on bitbang.

This time I disassembled all of OS9P1 and found where to put the patch. After a little experimentation I got it right and uploaded it to Delphi. Somebody pointed out that there was also a patch to OS9P1 that allowed OSK type file names, so I verified that my patch was compatible and re-wrote the docs.

So there it is, the complete history of the bitbang driver. Initially it was motivated by economics, then later by the desire to avoid having a multi-pak mess on my computer desk. It ain't pretty, but it does work.

Using Bitbang

BITBANG runs the CoCo 3 bit-banger serial port at 1200 baud under OS-9 Level II. If you have a telecom program, you can use XMODEM/YMODEM to download or upload files. It should even work with an external terminal (multi-user operations). Multi-tasking should be kept to a minimum while using bitbang, as the bit-banger port requires a good deal of processor overhead and will slow operations down considerably.

Installation

The distribution file should contain the following:

bitbang.doc - this file
bitbang.src - assembler source for bitbang
bitbang.dr - the assembled driver
newt1.src - assembler source for the newt1 device descriptor
newt1.dd - assembled newt1
newt2.src - assembler source for the newt2 device descriptor
newt2.dd - assembled newt2
maknewp1 - program to install patch on OS9P1

Source files and assembled code is available on microdisk or from the author (see sidebar).

There is a patch to the kernel that must first be installed. This changes the interrupt handling routine in OS9P1 to properly deal with FIRQs (it re-maps them as IRQs). Previously any FIRQ would cause a system crash. Even if you aren't going to use the bitbang driver this patch is a good thing to install for system stability purposes. The patch is installed "hot" - in memory while the system is running. This will not cause any problems as long as all disk access has ceased prior to installation; disk accesses cause NMIs which could cause OS9P1 to be entered in the middle of the patch process.

There are two versions of OS9P1 out there: the original and one that has been patched to recognize 68000 type file names. Their ids are:

Original	68000 type file name patch
Header for: OS9p1	Header for: OS9p1
Module size: \$0ED9 #3801	Module size: \$0ED9 #3801
Module CRC: \$969A94	Module CRC: \$C21516
(good)	(good)
Hdr parity: \$27	Hdr parity: \$27
Edition: \$10 #16	Edition: \$10 #16
Ty/La At/Rv: \$C0 \$88	Ty/La At/Rv: \$C0 \$88
System mod, Data, re-en, R/O	

The patch can be installed on either. To install the patch, ensure the execution attributes are set on "maknewp1" (attr maknewp1 e pe). Copy maknewp1 into your current

commands directory and load it into memory. After it is loaded and the disk drive light is out, install the patch by typing "maknewp1". The patch is now installed; you can use "ident -m os9p1" to check the results:

Original + FIRQ patch	68000 filenames + FIRQ patch
Header for: OS9p1	Header for: OS9p1
Module size: \$0EDA #3802	Module size: \$0EDA #3802
Module CRC: \$BBB284	Module CRC: \$B4250D
(good)	(good)
Hdr parity: \$25	Hdr parity: \$25
Edition: \$13 #19	Edition: \$13 #19
Ty/La At/Rv: \$C0 \$89	Ty/La At/Rv: \$C0 \$89
System mod, Data, re-en, R/O	

The new OS9P1 can be placed on your boot disk using COBBLER.

The best way to install the driver so that you can use it is to make a new boot disk. Copy bitbang.dr, newt1.dd and newt2.dd into the modules directory of your config disk and run config and make a new boot disk. The kernel patch will automatically be placed on your new boot disk by os9gen.

If you do not wish to generate a new boot disk, use cobbler to add the patch to the current boot disk, merge bitbang, newt1 & newt2 together and load them in the startup file like this: load foo; in iz t1; in iz t2. The t1 and t2 descriptors are identical except for name. T1 is the traditional name for the bitbanger port, but some telecommunications software is coded to look for a descriptor called t2, which is normally the ACIAPAK. Both are included so you can use the bitbanger port as either t1 or t2. If you are using an RS-232 pak you should not use newt2.dd, as t2 is normally the RS-232 pak. Use the t2 descriptor designed for whatever RS-232 pak driver you've got.

A minor modification to the 4 pin DIN connector on the serial cable coming into the computer is required: open the connector, remove the wire going to pin 1, then put a jumper between pins 1 and 2. This provides a FIRQ interrupt for incoming serial data.

Program Description

The source code files are well commented, but the methods may seem obscure. The whole thing is based on using software timing loops to read the individual bits of serial data at the right times. The write routine checks for incoming data while writing. If it sees a start bit coming in, it branches to a "duplex" routine that continues to transmit the current byte AND read the incoming data. At other times, incoming data causes a FIRQ, which is converted into an IRQ by a small patch to the kernel. The FIRQhand routine then reads the data bits, assembles them into a byte and stores them in a buffer. The buffer is 4 K long. Immediately after reading a byte of incoming data, if at 1200 baud, a special "chain delay" routine is executed to look for another byte of data close on the heels of the previous byte.

This condition occurs in Xmodem, Ymodem, and any time system response time to FIRQ is too long to catch bytes to close together.

Using the Program

For the most part operation is transparent. The user will notice (especially at 1200 baud) "jerky" screen response. Due to the large amount of CPU time spent in timing loops, the computer slows down A LOT when data is coming in. When the incoming data pauses for a while, the terminal program will read all of the data from the input buffer very quickly and the screen will suddenly display the data. The input buffer is 4K long - over two pages of text or 16 Xmodem blocks - so it shouldn't overflow. If the buffer does fill up, incoming characters are ignored. If possible, the user should set the host system to prompt for an input after each page of data is displayed.

For downloading or uploading, best results are usually obtained using the old "Xmodem" method; Xmodem-1K, Ymodem & Zmodem will usually require disk access in the middle of a block, which will cause a non-maskable interrupt (NMI). This really screws up the timing loops in bitbang and will cause aborted or timed out downloads. If you have a hard disk, RAM disk or a no-halt floppy controller you may be able to use XMODEM-1K or YMODEM since those devices don't cause the NMI. Experimentation will show the best method; old Xmodem always works (well almost ...).

For programmers, the following is a list of getstt / setstt calls that are recognized by the driver:

getstt:
 ss.ready - responds with notready error if buffer empty # bytes in buffer (if < 256)
 255 if >= 256 bytes in buffer
 ss.eof - never returns eof condition
 ss.scsiz - rows and columns from device descriptor
 ss.comst - port status (baud, parity, etc)
 setstt:
 ss.comst - sets new baud, parity, etc
 \$29 - Re-initialize the port and buffer

< 268'm >

The BITBANG programs, source code, and documentation are copyrighted by the author with all rights reserved. They may be copied and distributed for non-profit use only.

If you would like to encourage the author to write more programs, send a \$10.00 donation to:

Richard Kouke, Jr
 Rt 3 Box 431
 Crivitz, WI 54114

A disk copy of the latest revision will be returned. Any comments or questions about this program or its use may be directed to the author at the above address. Source code and executables are also on the 15 Sept. "microdisk".

If you just want OS-9...

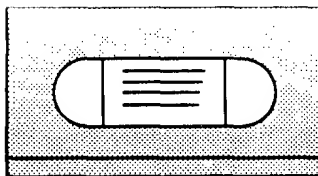


TO THE
 UNDERGROUND!

The "International"
 OS9 Underground
 Magazine

\$18.00/Year (12 Issues) U.S.
 (\$23. Canada, \$27. overseas)
 4650 Cahuenga Blvd., Ste 7
 Toluca Lake, CA 91602
 (818) 761-4135
 Write or call for Info

Video Tape Organizer



Jason Reighard

Organize your video or audio tape, even CD collection with this easy database!

"Video Tape Organizer" (VTO) requires a 32K CoCo but does take advantage of some CoCo 3 functions when run on a CoCo 3. It is a nice little program I originally wrote for my collection (more like a pile) of non labeled VCR tapes. I was tired of hearing "Do we have that on tape?" So now, by quickly loading VTO and using search, I can say "Yes it's on tape number 6 in drawer A.". VTO also prints lists (on standard paper) and labels (3"x7/8" address label size) for the tapes. It also alphabetizes your titles for added convenience.

The capacity of 300 entries can be expanded by changing the 300s in lines 30,1250,1330,2380 with the new total. Don't forget to clear more string space in line 30. A limit to the number will eventually be reached as memory runs out, so you may not be able to store any more info without creating more data files. This program should work well with something like "Big BASIC" or "512K BASIC" though. So lets keep track of those VCR tapes, cassettes, compact disks, etc.!

```
10 REM VTO
20 REM (C) BY JASON REIGHARD & FARNA SYSTEMS 1993
30 CLEAR 3000: DIM N$(300), T$(300), A$(300), S$(300)
40 IF PEEK(33021)=50 THEN WIDTH32: GOTO 50 ELSE 70
50 POKE65497,0:ON ERR GOT O 1940
60 ON BRK GOTO 2040
70 IF PEEK(&HC004)=214THEN N A$= "COEED52AD6D1D6F1D727D75E"ELSE A$="C101D617D7C4D7E4D81AD851":REM214=DECB1.0,215=1.1
80 FOR V=1TO 24 STEP 4
90 A=VAL("&H"+MID$(A$,V,4))
100 POKEA,189:POKEA+1,240:POKEA+2,157:POKEA+3,18
110 NEXT
120 POKE&HF09D,52:POKE&HF09E,127
130 POKE&HF09F,53:POKE&HFA0A,255
140 IFPEEK(&HC004)=215THEN N 160
150 POKE&HD6CD,0:POKE&HD723,20:GOTO 170
160 POKE&HD7C0,0:POKE&HD816,20
170 DN=PEEK(235):N=0:CLS:PRINT" VTO 1.1"
180 PRINT "BY JASON REIGHARD"
190 PRINT "COPYRIGHT 1993 JASON REIGHARD"
200 Z$="vTo":GOSUB 1870
210 Z$="vTo":GOSUB 1870
220 GOTO 200
230 CLS:PRINT"LOAD": PRINT@17,"DEFAULT DRIVE"DN:PRINT"VTO FILES ON DRIVE"DN:GOSUB2850:LINEINPUT"FILENAME >";F$:IF F$=""THEN350 ELSE IF PEEK(33021)=50 THEN 270 ELSE 240
240 IF LEN(F$)>8 THEN ER=1:GOTO1940 ELSE FOR X=1TO 8:IF MID$(F$,X,1)="/"ORMID$(F$,X,1)="" THEN ER=31:GOTO 1940
250 GOSUB 2730:IF OD=0 THEN ER=26:GOTO 1940
260 REM ****LOAD
270 OPEN"I",#1,F$+"/VTO"280 N=N+1
290 INPUT#1,N$(N),T$(N),A$(N),S$(N)
300 IF EOF(1)=-1 THEN 330
310 N=N+1
320 GOTO 290
330 CLOSE#1
340 REM ****MAIN MENU
350 CLS
360 PRINT"*MENU*":PRINT@17,"DEFAULT DRIVE"DN
370 PRINT"1 VIEW ENTRIES"
380 PRINT"2 EDIT ENTRIES"
390 PRINT"3 PRINT"
400 PRINT"4 LOAD"
410 PRINT"5 SAVE"
420 PRINT"6 DISK OPTIONS"
430 PRINT"7 ERASE ALL ENTRIES"
440 PRINT"8 ALPHABETIZE"
450 PRINT"9 END"
460 INPUT"SELECT1-9";M$
470 M=VAL(M$)
480 IF M<1 OR M>9THEN350
490 ON M GOSUB520,900,1360,2300,1770,2080,2400,2430,2040
500 GOTO 350
510 REM ****VIEW
520 CLS:PRINT"VIEW ENTRIES":PRINT
530 PRINT"1 VIEW ALL ENTRIES"
540 PRINT"2 SEARCH FOR A ENTRY"
550 PRINT"3 RETURN TO MAIN MENU"
560 INPUT"SELECT(1-3)";M$
570 M=VAL(M$)
580 IF M<1 OR M>3THEN520
590 ON M GOTO610,740, 00
600 RETURN
610 FOR X=1 TO N
620 CLS
630 PRINT N$(X)
640 PRINT"COUNTER"TS(X)
650 PRINT"TAPE#"AS(X)
660 PRINT
670 PRINT"PRESS ENTER TO CONTINUE":PRINT"PRESS X TO EXIT"
680 AS=INKEY$
690 IFAS=CHR$(13)THEN720
700 IFAS="X" THEN520
710 GOTO680
720 NEXT X
730 GOTO520
740 CLS:LINEINPUT"TITLE T O SEARCH FOR >>>";M$
750 IF M$=""THEN520
760 Q=LEN(M$)
770 FOR X=1 TO N
780 IF M$=LEFT$(N$(X),Q)THEN790 ELSE860
790 CLS
800 PRINT N$(X)
810 PRINT A$(X)
820 PRINT S$(X)
830 PRINT TS(X)
840 PRINT:PRINT"HIT ANY KEY TO CONTINUE"
850 IF INKEY$=""THEN850
860 NEXT X
870 CLS
880 PRINT@0,"NOTHING MORE UNDER THAT ENTRY":PRINT@32,"PRESS ANY KEY":AS=INKEY$:IFAS=""THEN880ELSE 520
890 REM **** EDIT
900 CLS:DE$="":PRINT"EDIT ENTRIES"
910 PRINT
920 PRINT"1 CHANGE ENTRY"
930 PRINT"2 ADD ENTRIES"
940 PRINT"3 DELETE ENTRY"
950 PRINT"4 RETURN TO MAIN MENU"
960 INPUT"SELECT(1-3)";M$
970 M=VAL(M$)
980 IF M<1 OR M>3THEN 900
990 ON M GOTO1010,1230,1070,1000
1000 RETURN
1010 CLS
1020 IAS=" CHANGE":AA$=" CHANGE":A$=" CHANGE":GOSUB 2550
1030 IF Z$="N" THEN 900
1040 IF DE$="N" THEN 900
1050 IF M$="" THEN 900
1060 GOTO 1140
1070 IAS=" DELETE":A$="ENTER TITLE YOU WISH TO DELETE":AA$=" DELETE":GOSUB 2550
1080 IF DE$="N" THEN 900
1090 IF Z$="N"THEN 900
1100 IF M$="" THEN 900
1110 INPUT"ARE YOU SURE (Y/N)??";M$:IF M$="Y"THEN
```

```

1120 ELSE 900
1120 N=N-1:FOR XX=X TO N:
N$(XX)=N$(XX+1):T$(XX)=T$(XX+1):
A$(XX)=A$(XX+1):S$(XX)=S$(XX+1):
NEXT XX
1130 GOTO 900
1140 M=X:CLS
1150 PRINT N$(M):PRINT"CO
UNTER "T$(M):PRINT "TAPE#
"A$(M):PRINT S$(M):PRINT
STRING$(32,"*");
1160 PRINT "PRESS [ENTER]
FOR NO CHANGE"
1170 LINEINPUT"TITLE>>>
";M$:IF M$<>""THEN N$(M)=
M$:N A$=MID$(N$(M),1,4):IF
N A$="THE " OR N A$="The "T
HEN1180ELSE1190
1180 LE=LEN(N$(M)):N$(M)=
RIGHT$(N$(M),LE-4)+",THE"
1190 LINEINPUT"COUNTER>>>
";M$:IF M$< >""THEN T$(M)=
M$
1200 T$(M)=M M$+T$(M)
1210 LINEINPUT"TAPE NUMBE
R>>> ";M$:IF M$<>"" THEN
A$(M)=M$
1220 GOTO 900
1230 CLS:PRINT "ADD ENTRI
ES"
1240 N=N+1:PRINT "REC."N
1250 IF N>300 THEN 2380
1260 PRINT"TYPE*END*WHEN
YOU DO NOT WISH TO ENTER
ANY MORE RECORDS"
1270 LINEINPUT"TITLE>>>
";N$(N)
1280 N A$=MID$(N$(N),1,4):
IF N A$="THE "OR N A$="The
"THEN 1290 ELSE 1300
1290 LE=LEN(N$(N)):N$(N)=
RIGHT$(N$(N),LE-4)+",THE"
1300 IF N$(N)="*END"THEN
N=N-1:RETURN
1310 LINEINPUT"COUNTER>>>
";T$(N)
1320 LINEINPUT"TAPE NUMBE
R >>> ";A$(N)
1330 IF N>=300THEN N=300:
RETURN
1340 GOTO 1230
1350 REM **** PRINT
1360 CLS:PRINT"PRINT"
1370 PRINT:PRINT"1 PRINT
ALL TITLES":PRINT"2 PRINT
TAPE LABEL":PRINT"3 CHANG
E BAUD RATE":PRINT"4 RETU
RN TO MAIN MENU"
1380 LINEINPUT"SELECT 1-3
";A$
1390 A=VAL(A$)
1400 IFA<1ORA>5THEN1360
1410 ON A GOTO1430,1510,1
640,350
1420 GOTO1360
1430 CLS
1440 LINEINPUT"MARGIN ";M
M$:M M=VAL(M M$)
1450 LL=VAL(LL$)
1460 PRINT#-2,TAB(MM);"TA
PE# COUNTER TITLE":CLS0
1470 FOR X=1 TO N

```

```

1480 PRINT#-2,TAB(MM);A$(
X);QP=LEN(A$(X)):PQ=7-QP
:GOSUB1490:PRINT#-2,T$(X)
:QP=LEN(T$(X)):PQ=9-QP:G
OSUB1490:PRINT#-2,N$(X):G
OTO1500
1490 FOR TA=1TOPQ:PRINT#-
2," ";NEXT TA:RETURN
1500 NEXT X:GOTO 1360
1510 CLS:LINEINPUT"TAPE N
UMBER YOU WISH TO PRINT "
:AC$
1520 CLS
1530 FOR X=1 TO N
1540 IFA$(X)=AC$THEN PRIN
T N$(X):AT=1
1550 NEXT X
1560 IF AT=1 THEN 1590
1570 PRINT"TAPE NOT FOUN
D":PRINT"PRESS (ENTER)"
1580 IF INKEY$=CHR$(13)TH
EN1360ELSE1580
1590 PRINT:LINEINPUT"IS T
HIS THE TAPE YOU WISH TO
PRINT? (Y/N)":AB$
1600 IF AB$="Y"THEN1610EL
SE IF AB$="N"THEN1360ELSE
1590
1610 LINEINPUT"MARGIN ";M
M$:M M=VAL(M M$):FOR X=1TON
1620 IFA$(X)=AC$THEN PRIN
T#-2,TAB(MM);T$(X);LE=LE
N(T$(X)):EL=7-LE:FORXX=1
TO EL:PRINT#-2," ";NEXT
XX:PRINT#-2,N$(X):NEXT X
ELSE NEXT X
1630 GOTO 1360
1640 CLS:PRINT"CHANGE BAU
D RATE"
1650 PRINT:PRINT"1 600":
PRINT"2 1200":PRINT"3 24
00":PRINT"4 4800":PRINT"
5 9600":PRINT"6 NO CHAN
GE"
1660 LINEINPUT"SELECT 1-8
";A$
1670 IF A$="1"THEN X=180:
GOTO1740
1680 IF A$="2"THEN X=87:G
OTO1740
1690 IF A$="3"THEN X=41:G
OTO1740
1700 IF A$="4"THEN X=18:G
OTO1740
1710 IF A$="5"THEN X=4:GO
TO1740
1720 IF A$="6"THEN1360
1730 GOTO1640
1740 POKE150,X
1750 GOTO1360
1760 REM **** SAVE
1770 CLS:PRINT"SAVE":PRIN
T@17,"DEFAULT DRIVE":LI
NEINPUT"SAVE FILENAME>> "
:FS:IF FS=""THEN350ELSE
IF PEEK(33021)=50THEN1790
ELSE1780
1780 IF LEN(FS)>8THEN ER=
31:GOTO1940 ELSE FOR X=1T
O8:IF MID$(FS,X,1)="/"OR
MID$(FS,X,1)=""THEN ER=3
1:GOTO1940

```

```

1790 GOSUB2730:IF OD=1 TH
EN PRINT"file already exi
sts":PRINT"OVERWRITE (Y/N
)"ELSE1820
1800 A$=INKEY$:IF A$="Y"TH
EN1820ELSE IF A$="N"THEN
350
1810 GOTO 1800
1820 OPEN"O",#1,F$+"VTO"
1830 FOR X=1 TO N
1840 WRITE#1,N$(X),T$(X),
A$(X),S$(X)
1850 NEXT X:CLOSE#1
1860 GOTO350
1870 FOR X=1 TO 20
1880 A$=INKEY$
1890 PRINT@12,Z$;
1900 IF A$=""THEN NEXT X:
RETURN
1910 GOTO350
1920 LL=VAL(LL$)
1930 REM **** ERROR MESSA
GES
1940 CLS:PRINT"THE FOLLOW
ING ERROR HAS OCCURED": R
INT:SOUND1,3
1950 IF ERNO=30THEN PRINT
"THIS DISK IS WRITE PROTE
CTED":PRINT"CHECK DISK":G
OTO2010
1960 IF ERNO=20THEN PRINT
"I/O ERROR":PRINT"CHECK D
ISK DRIVE":GOTO2010
1970 IF ERNO=31THEN PRINT
"BAD FILENAME":PRINT"YOUR
FILENAME CANNOT BE MORE
THAN EIGHT CHARACTERS
LONG AND CANNOT CONTAIN
A . OR A / ":GOTO2010
1980 IF ERNO=26THEN PRINT
"FILE NOT FOUND":GOTO2010
1990 IF ERNO=28THEN PRINT
"DISK FULL ":GOTO2010
2000 IF ERNO=6THEN PRINT
"OUT OF MEMORY ERROR":GOTO
2010ELSE PRINT"FATAL ERRO
R #" ERNO "IN LINE " ER
LIN
2010 PRINT:PRINT"PRESS
ANY KEY FOR MENU"
2020 A$=INKEY$:PRINT@14,"
ERROR":IF A$=""THEN2030E
LSE350
2030 A$=INKEY$:PRINT@14,"
error":IF A$=""THEN2020E
LSE350
2040 CLS:SOUND100,10:LINE
INPUT"ARE YOU SURE YOU WA
NT TO QUIT? (Y/N)":A$
2050 IF A$="Y"THEN POKE65
496,0:END
2060 GOTO350
2070 REM *** DISK OPTIONS
2080 CLS:PRINT@32," 1 CHA
NGE DEFAULT DRIVE"
2090 PRINT" 2 CHANGE STEP
RATE"
2100 PRINT" 3 RETURN TO M
ENU"
2110 INPUT D
2120 IF D<1ORD>3THEN2080
2130 IF D=1THEN2160

```

```

2140 IF D=2THEN2230
2150 IF D=3THEN350
2160 CLS:PRINT@17,"DEFAULT
T DRIVE" DN:PRINT"CHANGE
DEFAULT DRIVE":PRINT:PRIN
T"SELECT 0-3":PRINT"PRESS
[ENTER] FOR NO CHANGE":PR
INT:PRINT"CURRENT DRIVE N
UMBER"DN:PRINT:PRINT:PRIN
T:PRINT
2170 AS=INKEYS:IFAS=""THE
N2170
2180 IFAS=CHR$(13)THEN350
2190 IF VAL(AS)>5THEN2170
2200 DN=VAL(AS)
2210 IF DN>3THEN2170
2220 DRIVE DN:GOTO350
2230 CLS:PRINT@32,"SELECT
STEP RATE"
2240 PRINT" 1=6ms 2=
12ms"
2250 PRINT" 3=20ms 4=
30ms"
2260 PRINT:INPUT "Selecte
d Value";D$
2270 IF PEEK(&HC004)=215T
HEN2280ELSE2320
2280 IFD$="1"THEN POKE&HD
7C0,0:POKE&HD816,20:GOTO3
50
2290 IFD$="2"THEN POKE&HD
7C0,0:POKE&HD816,21:GOTO3
50
2300 IFD$="3"THEN POKE&HD
7C0,0:POKE&HD816,22:GOTO3
50
2310 IFD$="4"THEN POKE&HD
7C0,0:POKE&HD816,23:GOTO3
50
2320 IFD$="1"THENPOKE&HD6
CD,0:POKE&HD723,20:GOTO35
0
2330 IFD$="2"THENPOKE&HD6
CD,0:POKE&HD723,21:GOTO35
0
2340 IFD$="3"THENPOKE&HD6
CD,0:POKE&HD723,22:GOTO35
0
2350 IFD$="4"THENPOKE&HD6
CD,0:POKE&HD723,23:GOTO35
0
2360 GOTO2230
2370 IF PEEK(&HA282)=23TH
EN200:REM ADOS OR EADOS P
RESENT
2380 CLS:PRINT"RECORDS FU
LL!!!!":N=300:SOUND100,2
5:GOTO350
2390 REM **** QUIT END
2400 SOUND100,10:PRINT"AR
E YOU SURE ??? (Y/N)":IN
PUT AS:IF AS="Y"THEN2410E
LSE350
2410 QS="":FOR X=1TON$(
X)=QS:TS(X)=QS:AS(X)=QS:S
S(X)=QS:NEXT X:N=0:GOTO35
0
2420 REM***SELECTION SORT
2430 CLS0
2440 FOR Z=NT01STEP-1
2450 MX$=N$(1):MX=1
2460 FOR ZZ=1TOZ

```

```

2470 IF N$(ZZ)>MX$THENMX$
=N$(ZZ):MX=ZZ
2480 NEXT ZZ
2490 TMS=N$(Z):N$(Z)=MX$:
N$(MX)=TMS
2500 TMS=TS(Z):TS(Z)=T$(M
X):TS(MX)=TMS
2510 TMS=AS(Z):AS(Z)=AS(M
X):AS(MX)=TMS
2520 NEXT Z
2530 RETURN
2540 REM **** CHANGE AND
DELETE
2550 CLS:PRINT"ENTER THE
TITLE YOU WISH TO"IA$::L
INEINPUT" >>> "; \M$
2560 IF M$=""THEN RETURN
2570 Q=LEN(M$)
2580 FOR X=1 TO N
2590 IF M$=LEFT$(N$(X),Q)
THEN2600ELSE2690
2600 CLS
2610 PRINT N$(X)
2620 PRINT"COUNTER "TS(X)
2630 PRINT" TAPE# "AS(X)
2640 PRINT:PRINT"IS THIS
THE ENTRY YOU WISH TO "AA
$(Y/N)?"
2650 Z$=INKEYS
2660 IFZ$="Y"THEN RETURN
2670 IFZ$="N"THEN2690
2680 GOTO2650
2690 NEXT X
2700 DE$="N"
2710 CLS
2720 PRINT@0,"NOTHING MOR
E UNDER THAT ENTRY":PRIN
T@32,"PRESS ANY KEY":AS=
INKEYS:IF AS=""THEN2720EL
SE RETURN
2730 OD=0:SC=3:F1$=F$
2740 IF LEN(F1$)<8THEN FO
R X=1TO(8LEN(F1$)):F1$=F1
$+" ":NEXT X
2750 F1$=F1$+"VTO"
2760 DSKIDN,17,SC,AS,B$
2770 IFAS=STRING$(128,255
)THEN2830
2780 PS=1
2790 IFMID$(AS,PS,11)=F1$
OR MID$(B$,PS,11)=F1$THEN
OD=1:GOTO2830
2800 PS=PS+32:IF NOT(PS>1
28)THEN2790
2810 SC=SC+1:IF SC>11THEN
2830
2820 GOTO 2760
2830 RETURN
2840 REM **** LOOK FOR VT
O FILES
2850 FOR SC=3TO11
2860 DSKIDN,17,SC,AS,B$
2870 IF AS=STRING$(128,15
5)THEN2930
2880 FOR PS=1TO97 STEP32
2890 IFMID$(AS,PS+8,3)="V
TO"THENPRINTMID$(AS,PS,8)
2900 IFMID$(B$,PS+8,3)="V
TO"THENPRINTMID$(B$,PS,8)
2910 NEXT PS
2920 NEXT SC
2930 RETURN

```

< 268'm >

Announcing the

4th Annual Atlanta CoCoFest

Northlake Holiday Inn

(I-285 exit 28, west on LaVista Rd.)

October 2 & 3, 1993

Show Hours:

Sat. Oct. 2; 10:00AM - 5:00PM

Sun. Oct. 3; 10:00AM - 3:00PM

Vendor Setup:

Fri. Oct. 1; 6:00PM - 11:00PM

Sat. Oct. 2; 8:30AM - 9:45AM

Admission:

\$5.00 (whole show)

Reservations:

Northlake Holiday Inn

(800)-465-4329

or (404)-938-1026

\$52/night, two people

Ask for CoCoFest rates!

Vendor Spaces:

\$35; Includes one 7' table, one
chairs, and two admissions.

Sponsored by:

Atlanta Computer Society

P.O. Box 80694

Atlanta, GA 30366

BBS: (404)-636-2991

Several folks have asked me about the use of error correction and data compression on telecommunications applications in general, and over Telenet and Tymnet to Delphi in particular. All modern 14,400 bps, and many modern 2400 bps modems support one or both of the two common protocols for error correction and data compression.

One protocol is that set by the European CCITT. The error correction protocol is called CCITT V.42, and the data compression protocol is called CCITT V.42bis. The naming of these protocols can be confusing. These designations refer to capabilities in the firmware *inside the modem*. They do *not* refer to the hardware protocol by which the data is being put on the line. However, the terms CCITT V.22bis, CCITT V.32, and CCITT V.32bis refer, respectively, to the hardware protocols for putting data on the phone line used for the current standards for transmission at 2400, 9600, and 14,400 bits per second, respectively.

All modems I have seen that support the CCITT type error and data compression also support the older MNP (Micronet Protocol) type of error and data compression. MNP levels 3 and 4 are error correction protocols. MNP level 5 and up refers to schemes for data compression. Telenet and Tymnet, when used at 2400 bps and faster, support MNP level 3 or 4 error correction. Neither supports data compression.

Speed Conversion & Handshaking

All modems that support these features have the capability of having your computer talk to the modem at a data rate faster than that which the modem is connected to the telephone line. This is needed, because if you use data compression, your modem may want to send to your computer data *faster* than it is receiving the information on the line, for it is decompressing that information, and thus expanding it. This is called "speed conversion". This means that, while your modem may be putting data on the phone line at 2400 bits per second, your computer is talking to it at 4800 or 9600 bps.

In order to work properly, your modem needs a means of telling your computer to stop sending data to it briefly so it can catch up, in case you send stuff to it faster than it can compress and transmit it. The same is true in the other direction. This is referred to as "handshaking".

Handshaking is typically done in one of two ways: hardware or software. The hardware approach to handshaking is *by far* the best way to go. In this approach, two lines on the RS-232 port are used... one by the modem to tell the computer when to pause, and the other by the computer to tell the modem when to pause. The two lines typically used are the RTS and CTS lines. Thus, to fully use data compression, you need to have a hardware RS-232 port that supports the RTS and CTS lines, *and* you need to have a terminal program that knows about RTS/CTS hardware handshaking, so that you can activate that feature.

An alternative to hardware handshaking is the XON/ XOFF software approach. XON / XOFF software handshaking works pretty well if you are dealing only with ASCII text data (7 bit data, and just a few control characters). However, because the XON and XOFF characters are, themselves, ASCII characters, this kind of handshaking is useless if you are dealing with 8 bit binary information.

Preparing your Modem

To utilize the MNP or CCITT features of a modem, you need to set up that modem for that kind of operation. Sadly, as of this writing, there is little standardization among makers of modems as to what commands to give to activate and adjust these properties of the modem. The term "Hayes Compatible" is meaningless in this area. For example: ALL "Hayes Compatible" modems can be made to dial "555-1212" with the command "AT DT 555-1212". However, consider: To set a "Hayes Compatible" Prometheus PROMODEM 9600 Plus to "auto reliable" MNP mode and enable RTS/CTS hardware handshaking you'd give it the command "AT *E1 *F3". But to set a Multitech Multimodem 224E in the same fashion, you'd give it the command "AT &E1 &E4". Thus, while many features of a "Hayes Compatible" modem are standardized and can be used without any specific information on the particular make, model, and brand of modem, use of error correction and data compression capabilities usually absolutely requires careful study of the owner manual for that particular modem.

The most important things to look for in setting up your modem include:

1. Set the modem for *verbose* responses (AT V1) and for *expanded* MNP / error

protocol responses. The command(s) for this will vary from modem to modem. The effect of this is to tell your modem to announce to you after it connects whether or not it is in a protocol mode. Different makes and models of modems give differing amounts of useful information as to what they are doing. Most will distinguish between no protocol and error correction (often referred to as "reliable") mode, though many *will not* distinguish between whether or not they have added data compression to error correction in their connect message. Some will tell you whether they are using MNP or CCITT protocols, others may not.

2. Set the modem for "auto reliable" operation. This typically is done using the command AT &E1 or AT *E1, but the command may be different on your modem. Some modems, including some Hayes *brand* modems, use various "S" registers to control some of these functions. "Auto Reliable" means that the modem will look at information it receives from the host modem at the time of its attempting to make a connection, and automatically decide whether or not to turn on its special features depending on whether or not the host modem can support them.

3. Set the modem for *hardware* CTS/RTS handshaking. In *most* cases, you should also set the modem for "Speed Conversion".

There are many other parameters that some other modems allow you to set. On some you can specifically enable MNP but not CCITT protocol, or vice versa. On many you can tell the modem to utilize error correction, but to *not* utilize data compression even if the host modem supports that. This last can be important, for with MNP data compression, occasionally the file will actually be *expanded* a little if the file you are trying to "compress" is already compressed using one or another of the various compression/archive utilities.

Practical Guidelines

On Telenet and Tymnet with Delphi, there's no use for Data Compression (CCITT V.42bis or MNP level 5 or higher) capability in the modem because those services currently don't support that in their dial-up nodes. Data compression is a nice convenience if you are on line interacting with a bulletin board, looking at one screen after another, for data com-

pression can increase the effective speed of ASCII character transmission two-fold or more, making new screens appear considerably quicker on your terminal as you negotiate around a BBS.

Data compression is currently of very limited value in downloading specific files, for these days most files of any substantial size that you might want to download are already compressed with one or another "archive" type program, and further compression is usually not possible with modem firmware. Indeed, as I noted, MNP 5 may slightly expand such files!

Error correction (MNP up through level 4 and/or CCITT V.42) is highly desirable in almost all situations. It will virtually eliminate extraneous characters caused by "line noise". Indeed, MNP or CCITT error correction works so very well that my standard test of a new modem with that feature is to connect to Delphi using Telenet and MNP level 4, then pick up a voice telephone that is on the modem line and *whistle* into the phone for a few seconds, then hang up the voice phone. If the MNP feature is working properly, *nothing* appears on the terminal screen, and after hanging up the voice handset I can continue where I left off typing to Delphi, without any problems. If MNP is *not* working, garbage will appear on screen and the connection may even break.

As a minor technical point, I might as well note that MNP error correction does do a *tiny amount* of data compression as a side effect of the way it is implemented. In MNP level 3 and 4, asynchronous data is converted to a synchronous data stream, and with Class Three a 2400 bps connection can yield a throughput of roughly 2600 bps (108% efficiency), and with MNP Level 4 the implementation of variable block size into the protocol yields a throughput of roughly 2900 bps (120% efficiency) with a 2400 bps hardware connection. This efficiency is, of course, typically attained *only* if you have a continuous stream of data, as in a "straight ASCII, no handshake" type of download, *and* if you have your computer talking to your modem at a higher bps rate than the rate at which the modem is connected to the phone line.

I wrote this article to include items of interest both to folks who knew nothing about modem protocol error correction and data compression, and to those who already use these capabilities. Hopefully, this article has given you an overview of the subject... a framework within which to ask more detailed questions and seek answers from appropriate references and "experts".

< 268'm >

Connecting a Modem to the CoCo

F.G. Swygert

All of the OS-9/68000 (OSK) machines have one or more true serial ports that support RTS/CTS hardware handshaking. The CoCo's four pin "bit-banger" port does not - it only supports send, receive, carrier detect, and signal ground - the bare minimum necessary signals for serial communications. Incidentally, this port gets its nickname from the fact that the 6809 processes serial data via software and has to handle every bit that passes through the port individually. This creates quite a bit of processor overhead and is why the port is usually not used by OS-9 aficionados.

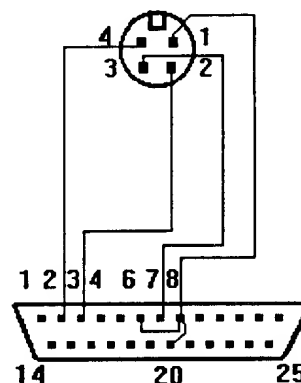
To use a data compression protocol with a CoCo a hardware RS-232 port will be needed. All the available ports support the CTS/RTS lines. The only currently manufactured ports are sold by Co-Nect. Other ports include the original Tandy RS-232 Pak, Orion Technologies Telepak, Disto RS-232 Super Pak, and a few older ones.

One important note is that the Disto multi-boards used with the Super Controllers can't be easily used with Disk BASIC programs. The RS-232 portion of these boards is addressed at The Tandy and work-alike paks are addressed from &HFF68 to &HFF71 while the Multi-Board port is addressed at &HFF54 to &HFF57. This is no problem for OS-9, but the address would have to be painstakingly changed within the DECB terminal program to allow operation.

In fact, a hardware RS-232 port is required for communications at *any* speed over 1200 bps under OS-9 and 2400 bps under Disk BASIC. To use the built-in port at all under OS-9 requires a special driver (see *Bitbang* in this issue). Some DECB software packages, such as Ultimaterm and Delphiterm, support up to 2400 bps through the serial port of the CoCo 3. The CoCo 1/2 port is only capable of reliable 1200 bps operation.

A cable for the RS-232 paks is easy - just buy any standard PC type 25 pin cable, or a

Looking from the outside at the serial port:



NOTE: The above diagram is for modems with DIP switch settings. For modems with no switches ("smart" modems), leave pins 6 and 20 unconnected and connect pin 4 to pin 8. This cable will not work with a printer. Printers are connected CoCo pin 2 to RS-232 pin 20, 3 to 7, and 4 to 3.

nine pin type if using one of the Co-Nect dual serial paks. Cables can be purchased with nine pin connectors on one end and 25 pins on the other in almost any gender combination, just pick the one that suits your needs.

Using the four pin port requires a special cable. If one can use a soldering iron, the cable is simple and inexpensive to make using the following diagram. Otherwise, order a cable from CoNect or get help from a user group or friend. I am unaware of anyone other than CoNect who advertises custom serial cables at a reasonable price. Rick will make up a six foot cable with any type ends desired for \$9.95 + \$4 shipping in US, +\$6 for Canada. See the *Bitbang* article for making a cable for the bitbang driver and OS-9.

< 268'm >

UPGRADE National Diskletter!

from Mid Iowa & Country CoCo. More than a local meet... we are a national reachout!

Our UPGRADE newsletter on disk is a real hit with Max III graphics accompanying articles. Articles range from Disk BASIC programming and hardware to OS-9 with news and editorials from across the country. Classified ads are FREE for subscribers. Print or read from screen.

Our Public Domain Library is purposely small. We feature only the "best of the best" in PD, shareware, OS-9, and big named Ophanware programs.

Our eight year record speaks very well to reaching out and serving members. A strong treasury assures confidence of delivery. We have members in 40+ states, Canada, and other countries. Our 90% renewal rate says most all feel MI&CC is very worthwhile. Join up and be part of the largest CoCo outreach in publication. Your membership brings you the UPGRADE newsletter disk, library, voting, and other privileges for one year.

ANNUAL MEMBERSHIP: \$16 US, \$21 Canada

Mid Iowa & Country CoCo (non-profit) "Gathering the CoCo Community"! Do you want support tomorrow? Stay with a group that keeps you in touch! Say "I saw it in 68' micros" and get an UPGRADE disk and free bonus disk!

Terry Simons (MI&CC Treasurer)
1328 48th Street

Des Moines, Iowa 50311
Phone 515-279-2576 after 8:00PM CST

Last month, we went on at length about the OS-9 System shipped by Tandy. Lots of dry reading, and the reader is no closer to using OS-9 than before! Well, it's going to happen one more time, while we cover the unofficial part of OS-9- what Tandy didn't sell you.

The first thing to check is your manual. There were at least two versions of the Level 2 manual put out by Tandy. They never told anyone, and there were no revision number changes to reflect the new manual. Look in the system commands section and see if the modpatch command has a page. If not, you have a very old Level 2 manual, riddled with bugs and omissions. I will send a list of corrections, including the missing text, on request.

In the years since Level 2 was released, 'hackers' have diligently attempted to improve every single byte of code in it. Tandy themselves contracted for a upgrade to Level 2, but cancelled the CoCo before it was finished. Due to certain legal problems relating to contracts with large multinational corporations, this upgrade is not yet available. However, work done before or outside the "Official" upgrade is fair game. Most of this work has been placed in the public domain, and is available from a variety of on-line services, bulletin boards, Internet sites, and so on. Those without use of a modem missed out on most of these neat patches previously, but no more!

For a small charge, every patch mentioned here can be obtained on a two disk set called "Patch_OS9". This set is available from either FARNA or CoNect. This set also has an auto-patch utility that will install the patches for you (if you have at least two double sided 40 track drives).

Some of the more common upgrades are listed below. There are many more available, this list merely reflects my personal likes. My main requirement is that everything be compatible, which eliminated some things that would otherwise be included. All of these are included on the "Patch_OS-9" disk set. FARNA also has a "Quick Reference Guide" that has short explanations of the standard commands AND the new commands in the disk set. Other items such as system calls and error codes are also in the QRG. It doesn't replace the manuals, but is easier to keep on your desk for a quick refresher on less used commands.

cc3disk: Modifies cc3disk to use any disk type the drive can physically handle. Includes two new commands (PCDOS and RSDOS) to read/write IBM or DECB disks.

clock : The edition9 clocks from Eddie Kuns clear up a nasty interrupt problem built into the CoCo 3. Formerly, this was fixed with hardware. If you spend much time on your modem, install this!

gshell+: Kent Myers has done a lot of work on the Multi-View system. This package changes everything! There is a shell hot key, a trash can to delete files or directories, you can double-click shell scripts to run them, enter parameters after double clicking a program or when it's AIF has ? as a parameter, list and print AIF files, and sort directories. The help system has it's own icon, so you can click for help. There are quite a few style changes to make Multi-View look better and all the bugs have been squashed.

grfdrv: Known as the "Christmas grfdrv", Kevin Darling's work really speeds up graphic functions. Some screens draw twice as fast!

kernal: Guy Loucks and Bruce Isted teamed up for this one. Allows OS-9 Level 2 to recognize up to 2 megabytes of RAM and accept OSK filenames.

os9p2 : There is a bug in OS-9's sleep call which can lock up the machine at times. This is the fix.

rammer: Mr. Darlings RAMdisk. It's size is not fixed, but can be set by the user up to 400K

rammer.pat: John Wesson fixed rammer to go to 768K (that's 96 tracks!)

rbf : released by Kevin Darling, this is another bit of the infamous Level 2 upgrade. Main claim to fame is the addition of the undelete command, which lets you restore any deleted files on a disk that haven't actually been written over yet. There is also a watchdog system call which alerts a process when somebody else tries to write to a disk drive.

sac1a: advanced serial port driver by Bruce Isted. Hardware flow control, bug fixes. This is the definitive 'stock' serial port driver nowadays.

scf : Another Kevin Darling patch, this allows editing the command line almost as though a word processor was connected to the OS-9: prompt. Real handy when doing multiple operations using the same pathlist or filename.

shell+: Ron Lammardo's shell+ fixes a lot of bugs... if that's all it did, it would still be a must have. But wait! You can put shell scripts in CMD5 or RAM, so they are always available. Useful prompts include the window #, proc id, std out, working dir, date, or time in the shell prompt. Better runb parameter passing. Command logging can save everything you have typed in to a disk file. Shell variables can store common command lines (or pieces of them.) Wildcards are supported. Redirection can add to or overwrite an existing file if desired. Shellscrips can have logic- IF/THEN/ELSE/ENDIF/GOTO/ONERRGOTO allow some pretty potent shellscrips. Path= lets you name additional CMD5 directories besides the main one. Security for multi-user systems is better. User startup files can be placed in any data directory. Some UNIX commands are recognized- cd cx | all work as well as chd chx and !. There is more, but this is getting pretty long!

In addition to the system level patches described above, there are tons of utilities, far to many to list them all. Some of the more useful ones (all on the "Patch_OS-9" disk set):

copy : This version from Mark Griffith is geared to Multi-View. Doesn't necessarily need a filename for between disk copies, or pathlists for copies to the same directory. If a file with the same name already exists, an Overwrite? window pops up.

dmode: Kevin Darling took a look at xmode and thought it would be real neat to change disk drive info the way you can change baud rates on the serial port... so he wrote this. Handy when one finds those weird 80 track, single sided drives, or just to use the last 5 tracks on a FD501. Rammer uses this utility to set the RAMdisk size.

free: gives sectors and bytes left. Who cares about blocks anyway?

kdutil : Kevin Darling wrote a collection of neat utilities, all packaged together here. proc adds the i/o paths to the procs output. DirM is an OSK style module directory. MMap graphically displays free RAM blocks. PMap shows what process is using which RAM. SMap displays system state RAM used and free. Lastly, DMem dumps sections of RAM to a file.

save : Yet another from Kevin Darling. Saves a module in memory as a disk file.

sdump : Marie-Louis Marcoux's screen dump will print the currently visible text window when shift-alt-ctrl is pressed. Ranks right up there with sliced bread.

wmode: Fred Sawtelle figured windows needed their own "mode" command. You can change the window descriptors to default to anything with wmode, and forget about wcreate or display commands.

This is nowhere near a complete list- just the stuff I use often. There are many other options out there, but be warned- there is no guarantee any given patch will be compatible with others.

All of the above can be installed at the same time, but be careful when expanding further. Sometimes the solution is to have more than one boot- for example, SCII owners will find that the non-CoCo disk patch and the SCII no-halt drivers conflict. Just make 2 boots- a no-halt OS-9 only version, and one that can write IBM disks.

Getting back to Tandy: Some things can be fixed without additional programs. The normal way to do this is changing the Operating system in RAM, then using cobbler to make a new boot disk. Cobbler copies what is in RAM to disk, so it saves any changes that have been made. It's fast and easy, but not perfect. The next time a new boot is made from the ground up, all those cobbled changes are lost, and must be redone. The best plan is to change the RAM, cobbler a new disk, then use save to copy the modules just changed to the modules directory you make boot disks from. Now the modifications are REALLY permanent.

First, the printer baud rate and serial baud rate can be set using xmode. The most common baud rate codes are 3=1200, 4=2400, 5=4800, 6=9600, and 7=19,200. Use xmode /p baud=# for printer and xmode /t2 baud=# for the serial port. Rename t2.dd and p.dd (in the modules dir) to xx.dd_tandy or something, and save the modified p and t2 descriptors just made, using the original names. Any future boots will have these changes included.

Wmode and dmode can be used to adjust window and drive stuff in RAM. Same routine as xmode- cobbler to make a permanent change on the current boot, save the descriptors for future boots so it doesn't have to be done again.

The modpatch command also changes modules in memory. What you do is type in a text file listing the changes, then

modpatch textfile to install them in RAM. Unless one of the above methods is used to save the changes, this has to be done every time OS-9 is booted. Where do you get the text files? Here are some:

80 COLUMN TERM SCREEN

```
l term
c 002c 28 50
c 0030 01 02
v
```

6MS DISK STEP RATE

```
l d0
c 14 00 03
v
l d1
c 14 00 03
v
```

6MS SPEED DURING BOOT

```
l boot
c 00c0 03 00
c 017c 13 10
v
l os9p1
c 05b7 cc 4f
c 05b8 00 5f
c 05b9 02 39
v
```

No overview of patching OS-9 would be complete without mentioning perhaps the biggest patch of all- 6309 native mode! This does require installing a new CPU in your CoCo- an Hitachi 6309E. This CPU adds lots of neat new hardware goodies, like a 32 bit accumulator, additional registers, and some neat new opcodes. It's also a CMOS chip, so it runs cooler and uses less power. The catch is somebody has to desolder the old, and add a socket for the new. Usually, that someone is you. Some computer repair shops will do this, although 'bench rates' can run pretty steep. CoNect can also install this chip, however you will be without a CoCo for a few weeks.

With the hardware part done, there are two variations of native mode available, Burke & Burke's Powerboost (\$24.95) and Gale Force's NitroS9 (\$34.50). Both companies also sell a 'kit' version, which includes CPU and socket for do it yourselves.

Powerboost ads claim an average speedup of 40%, which seems about right when comparing my current boot with the old. Although I haven't seen NitroS9 in action, rumor is it's slightly faster- either one gives you the equivalent of a 3Mhz machine. And that's average! Some operations appear to be 2 to 3

times faster.

One interesting side effect of the extra speed is the 'bitbanger' port quits working! Extra delay has to be added to the timing loops that drive this port, using the tuneport utility included with OS-9. Users have reported a delay of 99 to 116 is about right to get the printer going again.

tuneport /p -s=116

OK, that's it for now on the Operating System itself. Next issue, we get to the command line, and start putting OS-9 to work. As always, feel free to send questions or comments to me in care of this magazine, or by Delphi (RICKULAND), Internet (rickuland@delphi.com), or US Mail (449 S.90th, Milwaukee, WI 53214).

< 268'm >

OS-9 Users Group: Alive!

Many of you received notices that the OS-9 Users Group was being disbanded. This was the action of the president of the time and NOT supported by the group in general. The group has therefore reorganized with new officers. The new mailing address is:

OS-9 Users Group, Inc.
6158 W. 63rd St.
Suite 109
Chicago, IL 60638

The circumstances of the past months have made membership renewal a complicated issue. IF YOU WERE A MEMBER DURING THE MONTH OF JULY then your membership has been extended to the end of the year. You DON'T NEED to send in any money. We know who you are, so don't panic. IF YOU WERE NOT A MEMBER DURING THE MONTH OF JULY then you may become a member by sending in \$25.00 US/CA or \$30.00 foreign. Your membership will commence on that date (or soon thereafter).

MEETING ANNOUNCEMENTS

Meeting #1 - Reorganization Meeting
Date: Saturday, August 14th, 1993
Location: Chicago, IL (exact location TBA)
Open to: All OS-9 Users Group members
Subject: Rewriting charter and by-laws.

Meeting #2 - Annual Meeting
Date: Saturday, October 2nd, 1993
Location: Holiday Inn NorthLake; Atlanta, GA, 6:00pm (Coinciding with the Atlanta Fest)
Open to: All OS-9 Users Group members
Subject: New officer appointments. Minutes of the previous meeting will be presented, included a financial report. Open business will also be addressed.

Inquiries may be directed to the above address, CBJ@delphi.com, or boisy@microware.com.

Due to the great response I received from last month's article and sample program, I'm going to follow-up this month with some code that others have sent in to help improve upon the sample program.

First, Rick Ulland (Hi Rick, over there on pages 15-16!) mentioned an easier way to use the mouse values under BASIC09. In my program, I simply programmed in the offsets of the data I wanted from the mouse packet (such as the x- and y-values), and PEEKed into memory. Because the x- and y-values of the mouse are 2-byte integers, it was necessary to compute the mouse cursor data with the formula $msb*256+lsb$, where msb is the most-significant-byte (the 1st byte of the integer) and lsb is the least-significant-byte (the 2nd byte of the integer).

Rick pointed out a powerful feature under BASIC09... the TYPE statement. Using TYPE, you can set up an area of memory that can address a data structure (such as the mouse packet data) in it's correct form. So, using Rick's suggestion, the $msb*256+lsb$ stuff would turn into `Mouse.WRX`. I think you will all agree, this is much easier, and I'd like to thank Rick for sharing his insight! I don't believe this will work under OSK, since the mouse cursor data is still stored in 2-byte integers by the windowing system, but BASIC/OSK uses 4-byte integers. My thanks to Rick for sending me the following code:

```
PROCEDURE GetMouse
TYPE registers=cc,a,b,dp:BYTE; x,y,u:INTEGER
TYPE Mouse=Vld,Act,ToTm:BYTE;
X1:INTEGER; TTTto:BYTE; TSSt:INTEGER;
CBSA,CBSB,COACQ,ITSA,ITSB,ILSA,ILSB:BYTE;
X2,BDX,BDY:INTEGER; Stat,Res:BYTE;
AcX,AcY,WRX,WRX:INTEGER
DIM RegisterSet:registers
DIM callcode:BYTE
PARAM Pt:Mouse
RegisterSet.a:=0
RegisterSet.b:=$89
RegisterSet.x:=ADDR(Pt)
RegisterSet.y:=1
callcode:=$8D
RUN syscall(callcode,RegisterSet)
PRINT "mouse x:";Mouse.WRX
PRINT "mouse y:";Mouse.WRY
END
```

One of the best features of OS-9 is the signal. Signals allow, among other things, a program to put itself to sleep (so CPU time isn't wasted if there's nothing for the program to do), but awaken if the user presses a key or clicks the mouse button. If a keyboard button is pressed, one signal (value) is sent, but if the mouse button is clicked, another signal is sent, so a program using signals will be able to distinguish what event occurred while it was asleep.

So, signals are great, and under C, they work great, too. Unfortunately, BASIC does not provide any signal functions to the programmer! To me, this is a terrible oversight. So when Ted Jaeger wrote me how he managed to poke in a signal handler for BASIC, I was obviously interested. He

sent me several samples of source code using signals in various ways. I'm including two of them here. The first is a program for OSK users. Programmers may want to pay attention to how Ted sets up the OSK syscall registers, since I could not find the information on this in my BASIC manuals (I finally found out thanks to Chris Hawks).

The second program from Ted shows how to accomplish signal handling in BASIC09 on the CoCo 3. It is very similar to the sample program found in my column last month. My thanks to Ted for letting me include these in my column for people to learn from! As always, anyone with ideas, suggestions, or source code can send them to me care of this magazine or to my email address "JoelHegberg@delphi.com" on the internet (JOELHEGBERG@Delphi).

Listing of signals program for OSK users:

```
PROCEDURE osk_signals
(* Written by Ted Jaeger.
(* Aug 18, 1993
(* This program assigns signal to mouse
(* then sleeps for a bit
(* while asleep, user can click mouse
(* or press a key on the keyboard
(* to generate a signal.
(* The value of the signal will
(* appear in register d1 as can
(* be seen when register values
(* are displayed.
(* The registers are dumped in response
(* to any signal because an intercept routine
(* is stored in the BASIC variable icodeptcode
(* and run when a signal reaches the program.
(* KWindows push-buttons do not move
(* in and out since signals are in use.
```

```
BASE 1
(* define a variable for syscall
TYPE registers=d(8),a(8),pc:INTEGER
DIM regs:registers
```

```
(* define a variable to capture the mouse package
(* which is returned when the mouse is clicked
(* this is the large package with all mouse info
TYPE rodent=valid,area,ctrl,rsvd,msxmsd,
msxlsd,msymsd,msylsd,mssxmsd,mssxlsd,
mssymsd,mssylsd,mslftmsd,mslflsd,msmidmsd,
msmidlsd,msrgtmsd,msrglsd,mswxmsd,
mswxlsd,mswysd,mswylsd,msswxmsd,
msswxlsd,msswysd,msswylsd,mrv(12):BYTE
DIM msret:rodent
```

```
DIM F_Sleep:INTEGER
F_Sleep=$0a
DIM F_Icpt:INTEGER
F_Icpt=$09
```

```
DIM MouseSig:INTEGER
MouseSig=$0201
```

```
DIM KeySig:INTEGER
DIM key$:string[1]
KeySpec=$0202
```

```
(* poke in machine language code
(* to dump d0-d7 and a0-a6 to BASIC variable
(* the sequence of opcodes $48,$ff,$ff, etc are
(* the assembler signal handler
TYPE icodeptyp=stcode(4):BYTE; intaddr:
INTEGER; rcode(4):BYTE; intresult(16):
INTEGER
DIM icodeptyp:icodeptyp
icodeptyp.stcode(1)=$48
icodeptyp.stcode(2)=$f9
icodeptyp.stcode(3)=$ff
icodeptyp.stcode(4)=$ff
icodeptyp.intaddr=ADDR(icodeptyp.intresult)
icodeptyp.rcode(1)=$4e
icodeptyp.rcode(2)=$40
icodeptyp.rcode(3)=$00
icodeptyp.rcode(4)=$1e
```

```
(* must be defined as integer, not byte
DIM callcode:INTEGER
```

```
DIM signal,z,count:INTEGER
```

```
(* let's initialize the variable that will hold the
(* register values when they are dumped.
FOR z=1 TO 16
icodeptyp.intresult(z)=0
NEXT z
ON ERR GOTO 100
```

```
(* now we need to tell OSK that a signal handler
(* is installed. If OSK did not know, the program
(* would terminate on receipt of a signal.
callcode=F_Icpt
regs.a(1)=ADDR(icodeptyp)
regs.a(7)=ADDR(icodeptyp.intresult)
RUN syscall(callcode,regs)
```

```
(* screen pausing off
SHELL "tmode nopause"
count=0
```

```
PRINT CHR$(12)
PRINT "Click mouse or strike a key to generate a
signal!"
PRINT "I'll trap the signal and tell you what
happened."
```

```
(* enter loop where user can generate a signal
LOOP
(* initialize value in register d1
icodeptyp.intresult(2)=0
```

```
(* limit to 10 passes
count=count+1
EXIT IF count=10 THEN
ENDEXIT
```

```
(* give mouse a signal
regs.d(1)=0
regs.d(2)=$a4
regs.d(3)=MouseSig
callcode=$8e
RUN syscall(callcode,regs)
```

```
(* give keyboard a signal
regs.d(1)=0
regs.d(2)=$1a
regs.d(3)=KeySig
callcode=$8e
RUN syscall(callcode,regs)

(* sleep giving user a chance to generate the signal
callcode=F_Sleep
regs.d(1)=256*4
RUN syscall(callcode,regs)

(* the signal is returned in register d1 so print it
PRINT "Value in d1: "; iceptcode.intresult(2)
signal=iceptcode.intresult(2)
(* since it is known to BASIC, the signal can be
(* examined and the program can respond
accordingly
IF signal=MouseSig THEN
PRINT "You clicked the mouse!"
ELSE
IF signal=KeySig THEN
GET #0,key$
PRINT "You pressed the following key: ";key$
ENDIF
ENDIF

(* clear out the signals using _ss_release call
regs.d(1)=0
regs.d(2)=$1b
callcode=$8e
RUN syscall(callcode,regs)

ENDLOOP

END
```

```
100errnum=ERR
PRINT "Have error "; errnum
```

Listing for CoCo OS-9 users:

```
PROCEDURE CoCoMouseKey
(* Written by Ted Jaeger
(* Aug 19, 1993
(* For CoCo and BASIC09
```

```
(* read the mouse or keyboard via signals.
```

```
(* must be run on a graphics screen.
(* type 7 does nicely.
```

```
TYPE registers=cc,a,b,dp:BYTE;x,y,u:INTEGER
DIM regs:registers
```

```
(* Define a variable to hold the signal handler.
(* The signal handler is actually a short
(* machine language program that dumps the
(* contents of the b register to a BASIC09 variable.
(* The b register contains the signal when
(* sent by a device.
(* If no signal handler is present, the
(* program terminates upon receiving a signal.
```

```
TYPE intcpttyp=stbcode:BYTE;
intaddr:INTEGER;
rcode,intresult:BYTE
DIM iceptcode:intcpttyp
```

```
(* Now for the actual machine code
iceptcode.stbcode=$F7
iceptcode.intaddr=ADDR(iceptcode)+4
iceptcode.rucode=$3B
```

```
(* dimension a variable to capture mouse info
DIM mouse:STRING[32]
DIM a_button,sx,sy:BYTE
DIM callcode:BYTE
DIM key$:STRING[1]
```

```
(* text cursor off, echo off
SHELL "display 05 20"
SHELL "tmode -echo"
```

```
(* use hires mouse in right joystick port.
(* _ss_gip call
regs.a=0
regs.b=$94
regs.x=$0101
regs.y=$FFFF
callcode=$8E
RUN syscall(callcode,regs)
```

```
(* turn mouse on and tell pointer
(* to follow its movements.
(* _ss_mouse call
regs.a=0
regs.b=$89
regs.x=$030A
regs.y=1
callcode=$8E
RUN syscall(callcode,regs)
```

```
(* select standard mouse pointer
RUN gfx2("gcset",202,1)
```

```
(* Finally, tell os9 to invoke the signal
(* handler when a signal is returned.
(* Here we are telling os9 the address of the
(* signal handler routine and the location of
(* the variable holding the signal.
(* F_lcpt call
callcode=$09
regs.x=ADDR(iceptcode)
regs.u=ADDR(iceptcode)+4
RUN syscall(callcode,regs)
```

```
(* set up the screen
PRINT CHR$(12)
PRINT CHR$(2);CHR$(32);CHR$(32);"Press
<ENTER> to quit"
PRINT CHR$(2);CHR$(32);CHR$(35);"X value:
"
PRINT CHR$(2);CHR$(32);CHR$(36);"Y value:
"
PRINT CHR$(2);CHR$(32);CHR$(40);
"Keystroke: "
```

```
(* now enter the loop
REPEAT
```

```
(* need to initialize the variable that
(* holds the returned signal.
iceptcode.intresult=0
(* and the variable holding any keystroke
key$=""
```

```
(* tell mouse to return the value 10.
(* _ss_MsSig call
callcode=$8E
```

```
regs.a=0
regs.b=$8A
regs.x=10
RUN syscall(callcode,regs)
```

```
(* tell keyboard to return the value 8.
(* _ss_ssig call
callcode=$8E
regs.a=0
regs.b=$1A
regs.x=8
RUN syscall(callcode,regs)
```

```
(* now that each device has a signal
(* we can go to sleep.
callcode=$0A
regs.x=0
RUN syscall(callcode,regs)
```

```
(* when here, program woke up due to a signal
(* lets read the mouse if the signal was 10
(* or read the keyboard if the signal was 8.
IF iceptcode.intresult=10 THEN
callcode=$8D
regs.a=0
regs.b=$89
regs.x=ADDR(mouse)
regs.y=0
RUN syscall(callcode,regs)
a_button=PEEK(regs.x+8)
sx=PEEK(regs.x+24)*256+PEEK(regs.x+25)
sy=PEEK(regs.x+26)*256+PEEK(regs.x+27)
ELSE
IF iceptcode=8 THEN
GET #0,key$
ENDIF
ENDIF
```

```
(* release extra signal
callcode=$8E
regs.a=0
regs.b=$1B
RUN syscall(callcode,regs)
IF key$="" THEN
PRINT CHR$(2);CHR$(41);CHR$(35);" "
PRINT CHR$(2);CHR$(41);CHR$(35);sx
PRINT CHR$(2);CHR$(41);CHR$(36);" "
PRINT CHR$(2);CHR$(41);CHR$(36);sy
ELSE
PRINT CHR$(2);CHR$(43);CHR$(40);key$
ENDIF
```

```
UNTIL key$=CHR$(13)
```

```
(* Keyboard echo back on, and cursor on
SHELL "tmode echo"
SHELL "display 05 21"
```

<268'm>

Don't have a subscription yet?
WHAT ARE YOU WAITING FOR?!
Subscribe to
the world of
68'micros
TODAY!
 (details inside front cover)

Programming in "C"

Getting started: Input and Output

P.J. Ponzo

Editor's Note:

Well, I've had my first minor crisis as an editor. Mr. Sgambati was unable to write a column on C programming as he had planned due to unforeseen circumstances. This left me at somewhat of a loss. I had to find another author very fast!

Luckily, the telecommunications networks came to the rescue. The public domain files entitled "Lessons in C" will be printed here in the following months. The author graciously gives his permission for reprinting. Questions and problems may still be sent to "68' micros", as we do have someone to answer your "C" questions and help with problems.

F.G. Swygert, Editor

INPUT and OUTPUT

Using a word processor, we begin our C-program with main() and a left curly bracket {. Now we can write a program and finish it with a right curly bracket:

```
main() {
.....
your program goes in here
.....
}
```

Everything between { and } constitutes the main() program. One of the things we will want to do is print to the screen. There is a C command called printf, meaning "print with format".

```
printf("this is some text to print.");
```

The above command will simply print this is some text to print. The format part of printf comes in handy if we want to print, say, a number with a specified number of decimal places.

```
printf("the answer is %6d",x);
```

The above will print the value of the variable x.

```
printf("the answer is %6d",x);
```

This part just prints the words "the answer is".

```
printf("the answer is %6d",x);
```

This says the number x is printed to 6 digits.

```
printf("the answer is %6d",x);
```

This says that a decimal is being printed.

```
printf("the answer is %6d", x);
```

Here is the variable x which is to be printed. And (almost) every C statement ends in a semi-colon!

Here's the whole program:

```
main() {
x=1234
printf("the answer is %6d",x);
}
```

We expect the program to give the output: the answer is 1234... but it won't (yet). In fact, your C-compiler won't compile the program!

A look at Data Types

When we write x=1234 we expect the C compiler to reserve some memory for the variable x and place in this memory the number 1234. If, subsequently, we write x=12345678, will this fit into the memory reserved for the variable x? In order to know in advance, how much memory to set aside for each variable we use in our C program, we must declare the variable type.

If x is an integer (so we will *not* set x=1.234) then we say so with the declaration: `int x;`

If x will take on values like 1.234 (a floating point number) then we write: `float x;`

If x is a character variable (for example x='A') then we declare this too: `char x;`

Now we may write:

```
main() {
int x;
x=1234;
printf("the answer is %6d",x);
}
```

Now our program will print "the answer is 1234".

What will the following program print?

```
main() {
float x;
x=1234;
printf("the answer is %6d",x);
}
```

Because we declared x to be a floating point number the value of x is stored in a different format, in memory. As a float, x=1234 will be stored as a number *less than 1* together with an exponent ("scientific notation"). The above program will print "the answer is 0"!!!

```
float x;
printf("the answer is %6 d",x);
Note the "d"
```

The printf command expected a number in decimal format. But x was stored in memory in float format. The result is a misinterpretation of the value of x by printf!

```
main() {
float x;
x=1234;
printf("the answer is %6f",x);
Note the "f"
```

To tell printf the format in which x was stored, we use an "f" in the "format" portion of the printf command (i.e. the part within quotes).

```
printf("the answer is %6f",x);
```

```
main() {
char x;
x='A';
printf("As a character, x has the value %c",x);
}
```

What will this program print? As a character, x has the value A

Integers and Floats

```
float x;
x=1234+1/2;
```

In the above program segment, what do you

think the value of x is? It's 1234.0!!!

Numbers like 1234 and 1 and 2 are automatically integers and dividing the last two would give 0 (dropping the fractional part!). So 1234+0 gives 1234 which (because x is a floating point number) is now converted to a float and assigned to x!!!

In order to yield the value 1234.5 we include the decimal points:

```
x=1234+1/2.;
```

and now all numbers are floating point and x will be assigned the value 1234.5 so, include the decimal points when necessary!

```
float x;
```

```
x=1234+1/2.;
```

Here the 1./2. is evaluated as .5 (a float, because of the decimal points) and 1234 is converted to a float then added, giving 1234.5 which is assigned to x.

```
float x;
```

```
x=1234+1/2.; Note no decimal after 2
```

Here, the decimal associated with 1. causes the 2 to be converted to a float, so 1./2 gives .5, and the addition of 1234 (which is first converted to a float) gives 1234.5, which is then assigned to x.

The conclusion? C tends to convert numbers in a mixed expression to floating point before evaluationbut don't rely on it! If you want floating point operations, include the decimal points!

Input from the Keyboard with scanf()

Now that we've been introduced to printf (which will print things on the screen, or to the printer, or to a disk file, as we will see...) we should introduce scanf which allows us to input numbers 'n' such from the keyboard.

```
main() {
char x;
scanf("%c",x); Note the %c
printf("You pressed the %c key",x);
}
```

The scanf command waits for the user to type something, and scanf("%c",x) puts the character typed into the memory reserved for the variable x. When the compiled program is run it will wait for the user to press a key (followed by the "enter" key).

Suppose you ran the program and pressed the "a" key (then "enter"). What would the program print on the screen?

```
You pressed the key
ITDIDN'TWORK!!!
```

About Memory Locations...and the & prefix

```
main() {
char x;
scanf("%c",x);
printf("You pressed the %c key",x);
}
```

When scanf() is used, we must tell it where, in memory, to put the character x. Saying scanf("%c",x) is not enough! To identify the memory address for the variable x, we prefix x with &. If x is any variable (int or float or char) then &x is the address in memory of x.

The above program must be changed to read:

```
main() {
    char x;
    scanf("%c",&x);    Note the &x
    printf("You pressed the %c key",x);
}
```

...then it will wait for a key-press (for example the letter z) and print:

You pressed the z key

Since char x; reserved only enough space for one character, what would the printout be if you pressed several keys...say pqr?

You pressed the p key

JUST THE FIRST CHARACTER!

What's wrong with the following program?

```
main() {
    int x;
    char y;
    float z;
    printf("Enter 3 numbers (separated by
a space)");
    scanf("%d%c%f",&x,&y,&z);
    printf("x=%c y=%f z=%d",x,y,z);
}
```

Note that x, y and z are declared int, char and float.

printf("Enter 3 numbers (separated by a space)");

This line just prints instructions

scanf("%d%c%f",&x,&y,&z);

This line waits for you to type in the 3 numbers (with spaces). Note the &x, &y and &z as required by scanf()!

printf("x=%c y=%f z=%d",x,y,z);

Now we printf() the 3 numbers, expecting the printout to appear as: x=123 y=123 z=123 (assuming the 3 numbers were all 123). What do you think gets printed on the screen?

Alas, we gave printf() the data types in the wrong order! Although scanf() got the integer x as 123, and put it in the right memory location (because we said &x)...and it also got y as 1 (NOT 123), a single char... (because the y memory location only holds a single character!) and it also got z and stored it as a floating point number, and put it in the correct memory location... so what gets printed?

The printout produced by the above program (after entering 123 for each of x, y and z, separated by a space) is:

x={ y=-2.000000 z=16478

So printf() was confused once more! It interpreted x as a %character variable and out came { (the ASCII code for '{' is 123 ...but more on ASCII later). It printed the contents of several memory locations, starting with the one reserved for y (a char) as a %floating point number (and out came -2.000000). Finally, it went to the memory location reserved for z (which held a floating point number... in "scientific" format, remember?) and interpreted it as an integer, namely 16478!!!

Review of printf() and scanf()

We must give printf() information about the 'format' of the variables which are to be printed. Text may be mixed with the various %d, %c and %f format information.

printf("text and 'format' info", variable names separated by commas);

In here goes the variables to be printed (if any).

scanf("NO TEXT, just 'format' info", variable names separated by commas);

In here goes only format information for scanf(). In here goes the names of the variables, like x,y,z... right? WRONG! In here goes the addresses of the variables, like &x,&y,&z!

NOTE: These addresses are called pointers because they 'point' to the memory locations which are reserved for the variables.

Final comments on printf() and scanf()

We saw, earlier in this lesson, the statement:

printf("the answer is %6d",x);

We may specify the number of screen positions (the field width) which the number is to occupy when printf() is invoked (in the above example, it's 6 positions). For an integer, that's all that's necessary, but for a floating point number we may also specify the number of digits after the decimal. If we just say %f then we will get 6 decimal places (remember y=-2.000000?).

printf("the answer is %6.3f",x);

In this statement, if x were a float and equal to 12.3456 then what do you think would be printed? The answer is 12.346 rounded to 3 decimal places!

```
main() {
    float x;
    x=-12.3456;
    printf("x=%6.3",x);
}
```

Note that x=-12.3456 cannot be printed in 6 positions with 3 decimal places. The first 6 positions would give only x=-12.34 (or x=-12.35 if rounded). But printf() is a smart guy (gal?) and increases the field width as necessary and prints:

x=-12.346

For numbers (either int or float) the field width is understandable... but what about character variables which are single characters and will only occupy one screen position anyway?

x="Z"; What do you think
printf("x=%6c",x); this will print?

x=Z. the Z is right-justified in a field width of 6!

...and then there's Cast

Suppose you desperately need to have an integer variable converted to a floating point number. Maybe you wanted to compute the sine of an integer called x. Using sin(x) is no good because the sin(x) function only works for floating x.

You could use a "spare" floating variable, say float y; , and say:

y=x; (which does an automatic

conversion when it assigns the x-value to y, from integer to floating)

then use:

sin(y) (which is OK since y is now a float)

OR, rather than having to declare float y; just to have y in reserve for this "conversion" purpose, you may use the CAST.

sin((float)x) will temporarily convert the integer x to a float (for purposes of computing the sin) but leave x unchanged. This is called a cast (in C-speak).

int x;

x=10;

printf("Here's an integer, printed to 6 decimal places! %f", (float) x);

The above recasts the integer x for printing as a %f, and you get the output:

Here's an integer, printed to 6 decimal places! 10.000000

...and of course you may refer to (int)y to provide an integer copy of y (without changing y herself).

A final note on Data Types

We have talked about numbers stored (in memory) in integer and floating format. Whereas the integer format can represent an integer in the range -32768 to 32767 (with sign) and the floating format can hold a number with 6 significant figures (roughly) we also can declare a number to be double which is an extended floating point number and gives (about) 13 significant figures. Also, we can have an integer which runs from 0 to 65535 by declaring it to be an unsigned integer (which then is always non-negative). Although this type occupies the same memory space as integer, dropping the 'sign' allows for double the magnitude of integer. We can also have a long integer which occupies twice as much memory and can represent integers from (about) -2,000,000,000 to 2,000,000,000. The sizes given above, may differ from one machine to another!

The next lesson concerns strings and more formatting information.

P.J. Ponzo

Dept. of Applied Math

Univ. of Waterloo

Ontario N2L 3G1

<268'm>

CoCo-C

A complete Disk BASIC based C development package for the Color Computer NOT requiring the OS-9 Operating System. Contains:

Text Editor - A full featured line editor licensed from Bob van der Poel. CoCo 1,2, & 3 versions.

C Compiler - A full featured K&R style integer compiler. Assembly language output, position independent code, ROMable if desired. Can mix C, assembly, and BASIC within a program.

Assembler - Symbolic assembler capable of assembling files up to disk capacity. Supports Motorola syntax. Outputs files ready to LOADM and EXEC. List file output or symbol table.

Library/Linker - Utility that links CoCo-C's 90+ function library with compiled program, creating a standalone m/l file.

Command Coordinator - User friendly, menu driven interface that smoothly switches between the above programs.

Complete package includes CoCo 1,2, & 3 versions.

\$59.95+ \$4S&H (check, money order, or COD)

Infinitum Technology

P.O. Box 356

Saddle River, NJ 07458

Phone 914-356-7688

Repackaging the Color Computer

F.G. Swygert

The "CoCo3Go"... making the CoCo portable

I have had the idea of making a portable Color Computer for some time. I had seen a CoCo mounted in a portable Commodore 64 case (SX-64 Executive). This was a decent package, but the SX case is difficult to find (in 1985 the SX sold for \$900, whereas a C=64 was \$300, a CoCo 2 \$260). I then toyed with the idea of making my own case and using a 5" battery powered B&W TV for a display, but passed on that idea also. I finally arrived at the best solution: use an old "luggable" or "suitcase" PC type case. If I could locate one with an RGB monitor, that could be used with the CoCo 3. Although only nine colors would be displayed, 80 column text would be very clear. An adapter for monochrome screens is available from Barsoft (Box 281, Lake Villa, IL 60046; phone 708-587-9820), but it is not guaranteed to work with all displays. The 9" monitor can also be replaced with an inexpensive composite monitor (open frame, meaning no case) from Timeline (23605 Telo Ave., Torrance, CA 90505; phone 310-784-5488; call for availability). The luggable power supply could also be used.

The first suitable case I found, however, was not a PC case at all - it was a Kaypro II for \$25. This is really one of the best cases for a portable CoCo, I soon discovered. There is plenty of room inside and the conversion is rather simple. Not only that, but the monitor is a separate sync composite which can be used with the CoCo 3 RGB connector with an easy to make special cable devised by Marty Goodman. Other obsolete portable cases such as the Kaypro 4 or 10, Tandy Model 4P, Osborne 1 (5" screen), Osborne 2 (7" screen), StarLite, Jonos, Zorba (7" screen) and MicroStandard, will work as well as the Kaypro II. The majority of the CP/M (Z-80, all listed above) based machines will use some type of composite monitor, whereas most IBM compatibles use TTL monitors and will require an adapter or replacement monitor. All use nine inch monitors similar to the one available from Timeline except the Osbornes and Zorba.

A note of warning: stay away from LCD screens! I found a source for surplus LCD screens with a resolution of 640x200 (CGA/CC3 RGBA). These were very cheap and seemed ideal. The problem is that a special driver board is needed, and that would cost over \$200 even if I made my own! LCD computers usually have the drivers built into the motherboard, so those screens can't be used either. There is a chance that an LCD luggable or "lunchbox" has an LCD that uses a standard monitor port, but these are rare and

will be expensive. At one time Apple made an LCD panel for the IIc, but they were expensive way back when and are rare today (they plugged into the composite output of the IIc).

Back to the Kaypro II "upgrade"! Below is a step-by-step instructions for installing a CoCo 3 motherboard in the Kaypro II case. This should work for all Kaypro models as the cases are similar: There is room for a hard drive or battery pack (the drives and monitor would eat up lots of power though) and a 512K upgrade. The motherboard may have to be mounted a little lower to get a 2MB or Rocket upgrade board in due to the height of the SIMMs each require.

1) Remove the screws holding the cover in place. There are four on each side and two on the top.

2) Looking in the case from the back of the computer; the power connector will be found on the left side of the motherboard near the disk drives. Mark the voltages (silk screened on the motherboard) for future reference. Remove the floppy cable, reset /power on LED, and video connector (on the right rear corner in the Kaypro II).

3) Remove the motherboard. There are two screws near the front of the computer, to either side of the CRT and two on the back panel (removed from outside the computer). The posts on the serial port and screws on the parallel port also have to be removed.

4) There is a copper colored fiberglass groundplane attached to the bottom of the motherboard. Separate it from the motherboard, being careful to save the angle brackets that mount the groundplane to the back of the case.

5) Remove the disk drives. They were mounted with 7/64" allen screws in my Kaypro II. Mine was an early model with two full height drives. Only one can be used as the drive cage must be modified. Remove the drive cage now also.

6) Unless you know someone interested in the motherboard, it can be discarded. Before you throw it out though, pull the SYP-1793-002 (disk controller, used in long type CoCo controllers), MC1488 and 1489 (used in Tandy RS-232 Pak), and the 74LS243 (used in MPI). If you don't want them, push the pins in a small piece of aluminum foil wrapped styrofoam and send them to me. I'll send you a free issue of "microdisk" (your choice) for your trouble.

7) Remove the fiberglass support rods that held up the front of the motherboard. Shorten them 5/16" by cutting with a hack saw then reinstall. If they are a little shorter it won't

matter, but they must have at least 5/16" removed, 5/8" if using a 2MB upgrade or Rocket board. Also remove the reset button from the rear of the case.

8) Cut the back of the case out to fit the CoCo 3 motherboard as shown in figure 1. I discovered that the case is made of hardened aircraft grade aluminum. It is not as easy as one would suspect to cut (more like cutting thin steel than aluminum) and will break when bent. Using a 2MB or Rocket will require a bit more cutting as the existing holes won't line up as good. Also cut an opening in the front for the keyboard connector. I used a DB-25 connector mounted between the monitor and drives. Mine is near the bottom of the case, so I must unplug the keyboard when transporting. Mount about 1 1/2" from the bottom to keep keyboard attached. The keyboard may also be permanently attached by not using a connector. If doing this, make sure the ribbon cable is firmly anchored in the case to prevent detaching from the CoCo.

9) One of the angle brackets from the rear of the motherboard needs to be fastened to the right rear corner (looking in from the back of the case) of the ground plane. Drill a new mounting hole 5/16" below the original hole in the same corner (5/8" if using 2MB or Rocket).

10) A new support must be made for the left end of the motherboard and the disk controller. I used a 2"x6" piece of aluminum. I drilled it so that one of the parallel port mounting holes could be used and the cartridge connector can be screwed to it (fig. 2).

11) The disk drive cage must have the top rear corner removed. Cut 1" down and 1" back from the rear. This is required to clear the CoCo disk controller. The controller support will rest on this notch also.

12) Mount a half-height drive in the lower bay of the drive cage. If your drive has four mounting holes on each side, the upper holes must be used. If not, you will have to drill new holes for all drives. New holes must be drilled for the second drive anyway. Use a straight edge to draw a line between the original mounting holes then test fit the drives and mark the positions for the new holes. Kaypros with half height drives will have to have the front covers cut out for the drives also. Three half height drives or one of the full height and one half height can be used. The standard Kaypro II drives were single sided, though double sided drives were optional. All Kaypro 4 and 10 models used double sided drives, though early Kaypro 4s had full height drives. The Kaypro 10 has a full height 10MB hard drive and a half height double sided floppy. All

used 5.25" floppies. An SCSI hard drive could easily be used with a Disto Super Controller and hard drive adapter. One would have to be very creative to mount a Burke&Burke adapter, but it is possible. I taped pieces of heavy corrugated cardboard over the top drive and the rear of the open portion of the drive cage. This makes an ideal disk storage area.

13) Place the groundplane temporarily in the computer. Line the CoCo motherboard up with the holes in the back. Mark where the RGB connector needs to pass through the groundplane and cut a hole for the connector pins. I used a pair of shears and cut a notch from the side. Holes need to be drilled or punched where the cartridge connector and the 5V regulator are (fig. 3).

14) Once space for the RGB connector is cleared, put the CoCo motherboard temporarily in place and check the openings in the back of the case. Make any necessary corrections. Paint the case now if desired, as masking will be more difficult after assembly.

15) The Kaypro power supply must be used. The best way to connect a PC type power supply (or any external PS) to the CoCo is as follows:

a. Remove the screw holding the heat sink and 5V regulator to the CoCo 3 motherboard. Keep the screw, as it will be used to fasten the motherboard to the Kaypro groundplane.

b. Clip the 5V regulator from the motherboard. If you want to desolder it, clip the legs up high then remove individually. This is a common 5V regulator and can be easily replaced.

c. Connect +12V to the banded (cathode) ends of D2 and D14.

d. Connect -12V to the non-banded (anode) end of D4.

e. Connect +5V to the end of R19 that is closest to Q1 (the 5V regulator).

f. Connect ground to the keyboard ground clip land closest to the power connections or any other convenient ground on the motherboard.

NOTE: It is suggested that some type of four pin connector be used between the PS and motherboard to facilitate later removal if necessary.

16) After all corrections have been made to the case, reinstall the groundplane in its new location then place the CoCo 3 motherboard on the groundplane and fasten it in place with the heatsink/regulator screw and the cartridge connector screws. Place a spacer or washer about 1/16" thick between the cartridge connector and groundplane. This is necessary to allow easy insertion of the disk controller. I used 1/4" washers.

Now it's time for the video. Marty Goodman was kind enough to check the video on Wesley Ratcliff's Kaypro II for me. I extend my thanks to both of them! The video signal expected by the monitor is as follows:

Pin 1 - Horizontal Sync (TTL, +3V, upgoing, 15.75KHz)

Pin 2 - N/C (missing on motherboard, plugged on connector)

Pin 3 - Luminance (video data)

Pin 4 - Vertical Sync (TTL, +3V, downgoing, 60Hz)

Now the CoCo 3 doesn't put s out these signals, you just have to look for them! Connect the H and V sync from the CoCo 3 RGB connector to the H and V sync pins of the Kaypro II connector. Connect the anodes of three germanium diodes (1N34a) to the R, G, and B signals from the RGB connector. Hook all three of the diode cathodes to pin 3 of the monitor connector. This creates a reasonably clear image even in 80 column mode.

The keyboard connections were discussed in the last issue. If you mount the keyboard high for wrist support, you will have to notch the screen bezel for keyboard clearance when the case is buttoned up.

A finishing touch would be to remount the reset button from the back of the Kaypro II to the front and connect to the pins on top of the CoCo reset button. One may also want to connect the power on LED to +5V and ground through a 100 ohm, 1/4 watt resistor. My portable works great and will be at the Atlanta CoCoFest with FARNA Systems this year.

Next issue we will mount a CoCo 3 in a Tandy 2000 case and also discuss mounting in a PC case. I had a discussion with Rick Ulland of CoNect, and should have an Xpander to test by then. It will be nice to see how it mounts in a PC case.

< 268'm >

FIGURE 2:
CONTROLLER
SUPPORT

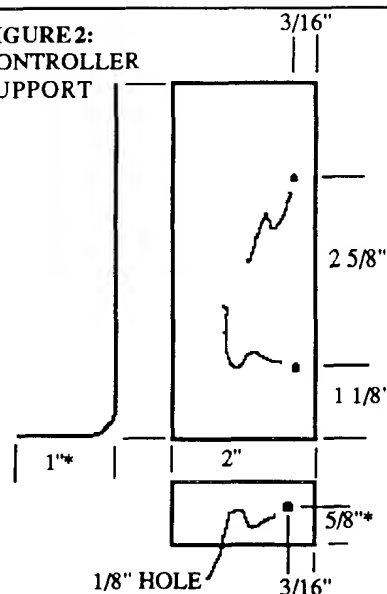


FIGURE 3: GROUND PLANE HOLES

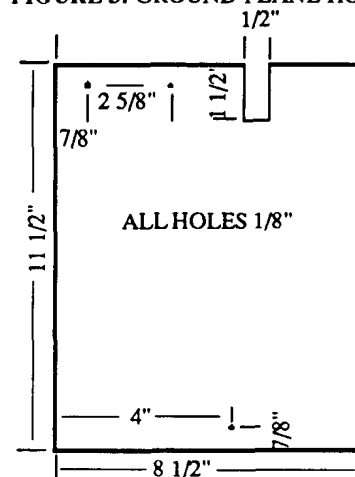
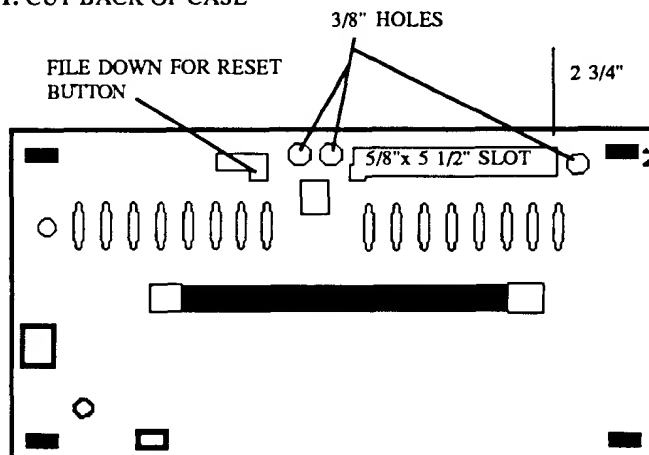


FIGURE 1: CUT BACK OF CASE



Beginner's Showcase

Programmers of all skill levels will appear here. The emphasis is on short, easy to type in programs that illustrate programming techniques. Typing in examples is a great learning tool! If you have a short program or subroutine in any language, drop us a line! Any program/subroutine printed may be used by anyone within their programs, even commercial programs, as long as credit is given the author and magazine within the code (REM statements) and documentation.

Repairing A Flaky CoCo

My CoCo started giving intermittent problems. Symptoms included freeze up during modem use, hard drive errors, floppy drive errors, RS-232 Pak problems, loss of DECB message after re-boot (ECB only), and periodic computer shut-down.

Removal of each component failed to eliminate all problems. I had added switches to my Disto SCII, but the connections failed to be the problem. Removing the MPI would not bring the computer back up after the problems started either.

My next thought was that the slot selection circuitry was failing. I had no info or schematics on the MPI, so asked for help on a local bulletin board. Within a few days I had info on the functions of the MPI, history of common repair problems, and a suggestion to check the power supply.

I put a scope on the +5V supply line and turned the system on. I read 5.03V. Then, the voltage suddenly dropped to 2.3V. I had found my problem! Every time the voltage in the MPI dropped, I would have problems.

I unplugged the MPI, put in all three devices (SCII, RS-232, and Ken-Ton HD controller) and turned it on. Within a few minutes, the +5V power dropped, but this time to 0V. Apparently, when the MPI supply cut out, the CoCo 3 supply tried to fill in, causing the main power supply to overheat and cut out. This is why the computer wouldn't return to normal after problems then removal of the MPI.

The power transistor, which is no longer available, turned out to be the problem. It did not require replacement though. The case is actually one of the leads for these transistors. The color and luster didn't look quite right, so I loosened the screws, swabbed de-oxidation solution (try alcohol, tuner cleaner, or other solvent type cleaners also) around it, then re-tightened the bolts. There was now stable voltage. Could it be that easy? After several weeks of use, the problem has not recurred.

Robert Gault

Editors Note: If you have simple suggestions like this, send them in! If it appears to be a common or hard to solve problem, we'll print it!

< 268'm >

This short utility is from Sam Thompson of Australia (reprinted from May/Jun 93 CoCo-Link). It calculates the "row, column" location when a PRINT@ value is entered. This is very useful when converting 32 column programs to 40 or 80 columns. It is suggested that the routine be loaded with high numbers as listed. Then one can start typing in or editing the new program. When a line with PRINT@ occurs, simply stop typing or hit BREAK and type GOTO 60000 and press enter. Change the multiplier in line 60020 to 2.5 and the 39s in line 60030 to 79s for 80 column screens. The values in the listing are for 40 columns. Maybe someone could reverse the program to create PRINT@s for locates.

Listing of "PRINTLCT.BAS"

```
59999 END
60000 INPUT"PRINT@ NO.";N
60010 IF N>511 THEN 60000
60020 H=INT((N-FIX(N/32)*32)*1.25+.5)
60030 IF H>39 THEN H=39
60040 V=INT((N/32*3/2)+.5)
60050 PRINT"USE LOCATE"H" , "V"
60060 N=0:H=0:V=0
```

< 268'm >

Fuel Consumption (FUELCONS.BAS)

Ted Beamish, another Australian, provides this automotive utility. It will figure your gas mileage and cost per mile or kilometer. Reprinted from May/Jun 93 CoCo-Link

```
10 REM FUEL CONSUMPTION          220 PRINT@206,"":INPUT
20 REM COPYRIGHT 1988            LC:GOSUB400:IFLC=0THENGOT
30 REM GUNGADIN SOFTWARE         O240ELSEGOTO230
35 REM BY T.BEAMISH              230 GC=LC*4.546:PRINT@213,
40 REM REPRINTED FROM MAY/JUN   USING"GAL=$$#.##";GC;:GO
93 COCO-LINK                     TO250
50 REM ALL RIGHTS RESERVED       240 IFLC=0THENPRINT@213,"
60 REM POKE359,57:SCREEN0,      GAL=":INPUTGC:GOSUB400:L
1                                C=GC*.219973603:PRINT@206,
70 CLS3:CLER:X$=CHR$(191:      USING"$ $$.##";LC;
PRINT@4,"GUNGADIN SOFTWARE      250 PRINT@224,STRING$(32,1
E(C)":SOUND123,1               79);
80 Z$=CHR$(128):A$=CHR$(19     260 MG=M/G:KG=K/G:ML=M/L:K
8):B$=CHR$(201):PRINT@33,A     L=K/L
$A$A$A$Z$Z"fuel"Z$Z$Z$Z$Z    270 PRINT@257,USING"mls/g
$"consumption"Z$B$B$B$B$;    ##.###";MG;
90 PRINT@65,"DISTANCE KLMS     PRINT@273,USING"klm/g
=MLS=                           al=##.###";KG;
";:PRINT@78,"":INPUTK:G       290 PRINT@352,STRING$(32,1
OSUB400:IFK=0GOTO160ELSEGO    88);
TO150                           300 PRINT@321,USING"mls/l
150 M=K*.6214:PRINT@85,USI     tr=##.###";ML;
NG"MLS=###.##";M;:GOTO170     310 PRINT@337,USING"klm/l
160 IFK=0THENPRINT@85,"ML     tr=##.###";KL;
S=":INPUTM:GOSUB400:K=M*      320 GOSUB400
1.609344:PRINT@78,USING"###. 330 IFGC=0THENGOTO340ELSEG
###.##";K;:IFM=0THEN90        OTO350
170 PRINT@129,"FUEL USED      340 INPUTQ:GOTO390
LTR= GAL=";                   350 Z=LC*L
180 PRINT@142,"":INPUTL      360 PRINT@449,USING"FUEL
:GOSUB400:IFL=0THENGOTO200    COSTS=$$#.##";Z;:ZM=Z
ELSEGOTO190                   /M
190 G=L*.219973603:PRINT@1    370 PRINT@431,USING"PER/M
49,USING"GAL=###.##";G;:     ILE=$$#.##";ZM;:ZK=Z/K
GOTO210                       380 PRINT@495,USING"PER/K
200 IFL=0THENPRINT@149,"G     LM=$$#.##";ZK;
AL=":INPUTG:GOSUB400:L=G      390 PRINT@302,"":INPUTQ
*4.546:PRINT@143,USING"###.  :GOTO70
##.##";L;:IFG=0THEN170       400 SOUND10,2:SOUND100,1:R
210 PRINT@193,"FUEL COST      ETURN
LTR=$ GAL=$ ";
```

< 268'm >

Telecom... Delphi, GENie, and Internet... David M. Graham & Allen Huffman

Rub the Lamp A. Huffman

A month has passed with the new \$3/hour rate in effect on GENie. Unfortunately, I don't have much to report as new in the CoCo/OS-9 section of the Tandy area there, but we have had a few uploads worth mentioning:

No. Name	Bytes
6056 ANSIFRONT.LZH	23808
Make 'C' compiler handle ANSI-C.	
6066 ETHADEMO.OSK.LZH	11776
Demo of new OSK user interface.	
6067 ETHADEMO.OS9.LZH	18304
Demo of new OS-9 user interface.	
6068 MULTIBOOT.AD	2688
Press release on new bootfile mgr.	
6070 CC250.LZH	23168
CC2.5.0 for use with ANSIFRONT.	

The ANSIFRONT/CC250 files will enable the Tandy/Microware K&R 'C' compiler to correctly process ANSI-C code. These files originated from Vaughn Cato (author of OSTerm and the classic bouncing ball demo). I believe we owe Vaughn a great deal of thanks for bringing the CoCo into "modern times" by giving us ANSI-C compatibility. These two files, along with replacement libraries (also available on GENie) will take the stock compiler and really soup it up.

The two EthaDemo files show a rough version of a new text user interface offering pull-down menus, pop-up dialog boxes, and mouse/keyboard support on high-speed, low-memory text screens.

The MultiBoot.AD is a Sub-Etha Software press release on a new OS9Boot utility which allows up to 16 bootfiles to be placed on one diskette with a scrolling menu to select which one to load after typing 'DOS'. These files are also available on Delphi and various bulletin boards and ftp sites around the country.

Speaking of Internet...

While GENie does not offer complete Internet connectivity, you can do many Internet functions. For instance, the Internet RoundTable (keyword "INTERNET-RT") has a menu option where you can request a program. They will search the Internet for that file then make it available in their software library for you to download. It's not instantaneous, but it does offer GENie subscribers full access to worldwide Internet ftp sites.

Internet mail access is also provided directly through GENie. Any subscriber now has an Internet mailing address. No special signup fee is required - Internet mail is billed just like normal GENie mail. A user on the Internet may address you as "GENIE-ID@GENIE.GEIS.COM". For in-

stance, my mail address is "COCO-SYSOP@GENIE.GEIS.COM" and Joel Hegberg's is "J.HEGBERG@GENIE.GEIS.COM". From GENie, to send a message out to the Internet you just type the TO: address and append "@INET#" on it. When I go to post a message to the CoCo ListServer, I send it to "COCO@PUCC.PRINCETON.EDU@INET#" and away it goes. I can even send mail to users on Delphi or CompuServe. This is one great advantage of the Internet - it offers a mutual mailpath for various competing online services...and it offers GENie CoCo users another gateway to use for files and support.

I'll try to go into more details on using Internet mail in a future installment of this article. Until then, feel free to drop me a line with any special comments or questions. I can be reached through this magazine or direct at: P.O. Box 152442, Lufkin, TX 75915. Or, you may also use my GENie Internet address!

On-Line Currents D. Graham

Hi, I'm back again! I'd like to talk this month about how online files are organized on Delphi.

Delphi offers a variety of services of course, including it's SIG's. The computer Special Interest Groups share a common menu directly off of Delphi's main menu, and most of these SIG's have their own database area as an option off of the SIG menu. I say most because it is possible for two SIG's to share a database with a related SIG - for example, the PC SIG and the Tandy SIG. Within the database area itself, the files are divided into storage areas by file type, according to the needs of that group.

Most groups have an area called NEWFILES or RECENT ARRIVALS. That area is what you will most often see me reporting on in Delphi's OS9 Online database. The CoCo SIG on the other hand, is one of the rare exceptions to that rule. I search each area of the CoCo Database each month, and hope to catch everything!

The OS9 Online Database selections (topics) are:

- General Information
- Users Group
- Applications (6809)
- Telecom (6809)
- System Modules (6809)
- Games & Graphics
- Music & Sound
- Programmers Den
- OSK Applications
- OSK Telecom
- OSK System Modules
- Rainbow OS-9 Material (\$)

Tutorials & Education
Standards
New Uploads

The CoCo SIG Database selections are:

- General Information
- CoCo 3 Graphics
- Source for 6809 Assemblers
- Utilities & Applications
- Hardware Hacking
- Games
- Classic Graphics
- Music & Sound
- Info on Rainbow
- Archives
- HELP
- Product Reviews & Announcements
- Rainbow On Tape
- Telecommunications
- Soapbox (chitchat)

Once you select a topic, you can do a directory of the files stored within the topic, and then 'READ' the ones you like. When you read a posting, you are given a chance to download the files.

OS9 Online New Files:
OXYGENE "MOD" MUSIC FILE
DISCUSSION ON BEZIER CURVES
AUSTRALIAN CONNECTION!
STGNETWORKBBS V3.0
VT100+ VT220 ADDITIONS
OS9FAQ ED. 8 OS9 Frequently Asked Questions
RSMENU 1.0: for DECB to OS-9 transfers
TR3:SEARCH/REPLACE 3 BYTE SEQUENCE
MERGE EDITION 5
OS9LIB.L: UNIX COMPATIBILITY LIB for OSK
UNPROTO—CONVERT ANSIC TO K&R
UMUSE3 6309/MULTIVUE PATCHES
STAR WARS ADVENTURE
FIND: UNIX STYLE FILE FINDER
DYNACALCTRM EXPLAINED
BLACKHAWK-HYPERTech ANNOUNCEMENT
CC2.5.0 FOR ANSIFRONT: 6809 C compiler executive, calls ANSIC front end.
STERM V1.5.1: SIMPLE TERM. for OS9/OSK systems
PEARLS FOR G-WINDOWS
UNARJ 2.21 FOR OSK:
Uncompress MS Dos ARJ archives
MULTIBOOT PRESS RELEASE: from SubEtha Software
C_PREP V1.6 ANSIC PREPROCESSOR For CoCo 3
ACIADRV SERIAL DRIVER
JEFFERY DAHMER VF2
STICKY: OSK modules stay in memory!

GCC V2.4.5: GNU C COMPILER for OSK

New DECB Files:
REPORT ON THE FLOOD
GIF87A/GIF89A FORMATS SPECS
EDT/ASM 6309 BUGS
EDT/ASM 6309 DISASSEMBLY
Use EDTASM for 6309 systems
ETHASAMPLE DEMO

Internet FTP David & Allen

The Internet offers many different services. ARCHIE, FTP, LISTSERV, GOPHER, and much more, but the one that concerns us most at this point is FTP.

FTP stands for File Transfer Protocol. This offers you a chance to sign on to a computer system, just as though you were at the keyboard, and transfer files to your system, or another system you may be signed in on. This means that you can sign on to a commercial net and then use that net's computer to sign on to another computer, theoretically ad infinitum.

An FTP address looks like any other Internet address, and is a 'handle' you use to tell the FTP program where to call. Internet addresses can tell you quite a bit if you know the layout. The first part designates the user ID of the individual or program you are addressing. The second part describes the computer system you are calling and the type of organization that operates it. The two parts are divided by an ampersand (@). 'delphi.com' states that the Internet site you are calling is named delphi and it is a COMmercial system. Other systems may have more than one name, and may be GOVERNment, EDUCational, or MILitary.

The multiple names tell that network's Internet servers which local computer you want to access. OS-9's primary FTP storage site in North America is called cabrales.cs.wisc.edu. Note the lack of an '@' sign. FTP programs are computer names only. Cabrales is an individual site name, cs presumably stands for Computer Science, and wisc stands for WISconsin.

When your software prompts for the desired FTP site, you just enter the site name. Next you enter your login name. For most FTP packages, you use 'anonymous'. This allows the general public to have a common means of access to unclassified files. For the password, use your username at your local site (the last system you signed on before you called the FTP site). Once logged on, what you find varies with the system called.

We'll discuss more about Internet in the next issue!

<268m>

Blackhawk Enterprises has announced a license agreement with REMCOMS of Australia to manufacture and sell MM/1s. Plans are to sell the MM/1 in Australia and neighboring countries, including some in Europe.

In order to reduce delivery time, Blackhawk has been making an effort to keep several MM/1s in stock. They are making an effort to keep as many in stock as business allows. IMS tries to work through zero inventory, building computers as they are ordered. This method might work for them, but it sure doesn't sit well when the customer has to wait 90 days or more for an expensive product! Blackhawk is attempting to remedy this problem. If you are looking for an MM/1, try contacting them (see ad in this issue).

Frank Hogg Labs has announced a special deal for software developers. If you have a package that FHL is interested in, and you are willing to grant FHL exclusive rights to sell it (they will pay royalties), then you can get a complete G-Windows system at a whopping 66% discount. If you don't want to grant FHL exclusive rights, but will grant them selling

rights, you can still get 50% off. There are some restrictions, one being that FHL has to see merit in your idea, another being it has to be developed under (or ported to) G-Windows. There are others... write FHL for details.

DISTO is considering building a new batch of 4-N-1 boards. These boards fit into their Super Controllers and have a real time clock, serial port, parallel port, and SASI/SCSI hard disk interface. It can be used in an MPI with a special adapter. Contact DISTO if you might be interested. The boards should run around \$130.

Micro News....

More news from Blackhawk! They have a port of K-Windows in the works for the Delmar System IV and System V. The port isn't expected to be completed until mid 1994 though. This will enable software written for the MM/1 under K-Windows to run on the Delmar machines.

Is everyone ready for the 4th Annual Atlanta CoCoFest? It will be held at the Northlake Holiday Inn of Atlanta on the

2nd and 3rd of October. See the ad in this issue for details. For those who don't know how to get there, take I-285 until you get to exit #28 (LaVista Road). This is in the North East quarter of Atlanta, splitting the city with I-75 and I-20. When you leave the Interstate, turn West (toward downtown Atlanta). The entrance to the Holiday Inn is to the right, so get in the right lane. If memory serves well enough, the entrance is less than 1/4 mile off the Interstate, so watch for the sign as soon as you turn onto LaVista Road. FARNA Systems will have a booth and back issues will be available.

For those whose subscriptions didn't arrive in my office before 15 August and requested first issues, I apologize. I simply can't start everyone with the first issue! I sent the first issue out 15 July and continued until 15 August... when I ran out of the *third* batch! At this rate our goal of over 200 subscribers by January 94 will be easily exceeded. For those who still want a first issue, I will waive the \$1 shipping fee and reduce the price to \$3.50. This applies *only* to those with subscription end dates of 9/94 (on your mailing label).

< 268'm >

If you have new soft or hardware products, let us know! We will gladly print a free blurb for you here in MicroNews whether you advertise or not (though we will be happy to have your ad also).

The Swap Shop

Buy, sell, or trade hardware and software!

If you have something you no longer need, need something you can't find, or just have some extra "stuff" lying around, this is the place to find or get rid of it! Sorry, no complete systems advertised.

Subscribers receive ads free. All others pay \$0.20/word with a \$4.00 minimum. Name, address, and phone number are free. Only those vendors with at least a 1/4 page ad may place classifieds without special permission. For these, short ads will be free as space permits.

FORSALE

64K ECB CoCo 2 - \$20. Jason Reighard, 441 Ridgeland Dr., Toronto, OH 43964

Multi-Pak, 26-3024 (large): Slot switching circuit needs repair, can supply instructions. \$45 includes shipping. F.G. Swygert, 904 2nd Ave., RAFA, GA 31098-1029. 912-328-7859 (8-11am EST wk days)

Multi-Pak, 26-3124 (small), upgraded for CC3- \$70; OS-9 Level II- \$35; Multi-Vue-\$10; Koronis Rift-\$10; Rogue-\$10. Shipping included to US. John Gar, RR1 Box 141, Newel, SD 57760

SALE OR TRADE

I have approximately 24 never used Tandy CC3 programs for sale or trade for CC3 soft/hardware. Send SASE for list. John R. Mott, Jr., Box 26246, Phoenix, AZ 85068-6246

Extra CoCo 3s! Have 5 CoCo 3s, 6 CM-11 monitors, 3 5 1/4" drive systems. Call and make offers on one or all. Arnold Stark 813-654-4198 (day) or 813-621-4987 (evenings)

CoCo 2, 64K; CoCo 3, 128K; CM-8 monitor, dual 5 1/4" floppy system. Call and make an offer! Jenny 510-779-1102

TRADE No Entries

WANTED

Used CoCo hard/software. Will buy by piece or entire collections. Rick Ulland, 449 South 90th St., West Allis, WI 53214

E.A.R.S. hardware/software by Speech Systems. Call Johnny at 713-479-4002

NONCOMPUTER RELATED

Do you need to add a professional look to your software manuals? Need inexpensive typesetting? Publishing your own book? Prices \$5-\$15 per page! We typeset masters for your printer or will print for you. Send a description of what you need to FARNA Systems PB, Box 321, Warner Robins, GA 31099-0321. Phone 912-328-7859 (8-11am, also weekends).

NOTICE: Free personal classifieds will be printed for those with E-mail access on a space available basis. Send to DSRTFOX on Delphi, dsrtfox@delphi.com on Internet.

DISTO Products

Hardware for your CoCo!

2MB Upgrade (no RAM) - \$99.95

Mini Disk Controller - \$70

Super Controller I - \$100

Super Controller II - \$130

MPROM Burner - \$50

3-N-1 (parallel, RS-232 RTC) - \$75

SASI/SCSI Hard Disk Adapter - \$75

Full Turn of the Screw - book

with Rainbow articles - \$20

Complete Schematic Set - all

DISTO schematics except 2MB upgrade - \$20

Include \$2 S&H for book or schematics. Hardware S&H is \$4.50 for single item, \$6.50 two or more.

Certified check or International Money Order only!

DISTO

1710 DePatie,
St. Laurent, QC H4L 4A8
CANADA

Reviews...

Supercomm

Supercomm is a telecommunications package for the Color Computer 3 running under OS-9 Level II. This program was originally sold as a commercial software package, but has since turned into public domain. Randy K. Wilson (RANDYK-WILSON on Delphi) now has the source code and does all updates and bug fixes.

Some of Supercomm's features include ANSI-BBS terminal emulation, an autodial feature with a max of 30 entries, 10 keyboard macros which can be set up differently for each entry in the autodial directory, on-line timer, key-click, ASCII capture buffer, on-line help, conference mode, shell access and file transfers.

The file transfers are probably some of the best parts of Supercomm. Xmodem, Xmodem-1K, Ymodem batch and Zmodem (with supporting sz and rz programs) are supported. A nice thing about the transfers is that Supercomm will do reliable 9600 baud downloads although multitasking is seriously hindered at that speed.

Setting up Supercomm is a relatively easy task, the only thing you really need to do is create a directory on the default drive for storage of the autodial entries. Besides just dialing the modem the autodial entries also can be used to change communication parameters such as baud rate and parity, load in keyboard macros, and auto-logon to the computer system being called. For the on-line timer, you must install Bruce Isteds VRN and /nil package in your Os9boot. This change will also increase your Xmodem and Ymodem throughput.

The only bug I could find was when you open a capture buffer on start-up the program bombs. Randy has been notified of this problem and it will be corrected in the next release.

Overall I think most users will be happy with this terminal program as it has most of the wanted features in such a package, 6309 support, and the price is definitely right as this program can be downloaded from Delphi or GENie. Personally, I think the docs could be improved a little bit, but they do explain all of the program's features adequately.

Quinn Granfor

Editor's note: Supercomm is included on this issue's "microdisk".

OS-9 Disk Mailing List

As the name suggests, this is a mailing label program for OS-9. Versions are available for CoCo OS-9 and OS-9/68000. The CoCo version was tested, but the 68000 version is similar.

DML9 is easy to set up. Simply copy the main DML9 program into the CMDS directory. Since DML9 is in Basic09, RunB must be available also, either in the same CMDS directory or in memory. A data disk is then prepared. This needs to have the .DMLmask files copied to it. I prepared a single disk with all necessary files. All data files created by DML9 will have a .DMLdata extension. The program will run in a 32 column window (CoCo version will run under level I), but at least a 40 column, 16 row window is needed to properly display all prompts.

A mask file defines the format labels will be printed in. The file consists of a string of data containing the print parameters. Parameters include fields to print, format to use, and constants. Three ready made masks are included - address labels, an address book listing, and a phone listing.

Data fields are fixed. The field names are: FNAME, LNAME, ADDRESS, CITY, STATE, COUNTRY, ZIP, PHONE, MEMO. The additional fields CDATE (created date) and PDATE (last printed date) are entered by the program and cannot be changed by the user. They can, however, be used as printing variables (one can print a specified created date, etc.) and can be used in the mask file.

Files can be sorted in two ways - by name or zip code then name. The program defaults to name order by last name. The zip code order takes longer to sort, but bulk mailers will find it very convenient.

The program is menu driven and easy to use. Even if there is a major error, the error routines close all files before ending the program.

I did find a few features a little hard to get used to, but I attribute that to the fact that I am just starting to use OS-9 regularly and not to the program. In every case, referring to the documentation corrected my difficulties.

With the flexibility of this program, it is a must have even if one just keeps a personal address/phone list and doesn't run a business. DML9 is used to mail "68" micros". Cost is only \$24.95 + \$2 S&H.

F.G. Swygert

Bob van der Poel Software
Box 355 Box 57
Porthill, ID Wynndel, B.C.
83853-0355 Canada V0B 2N0

This Issue of "microdisk"

For those with OSK machines, there is a special disk with the free-ware telecommunications program "S-term". This particular version has been modified to support Z-modem transfers. The author does not support this modified version since it was made by another individual.

If you already have a subscription to "microdisk", drop us a post card with your expiration date (after your name) and request a free OSK disk. Don't forget to mention the issue date! If ordering a single issue, specify the date and OSK. The standard "microdisk" AND the OSK version may be ordered together for only \$2 over normal single issue price (\$8). Note that the OSK version is still CoCo format, but double sided 40 track (5.25").

A Note on ".AR" and ".ARC" Files

One will notice that almost all the OS-9 files have an ".AR" extension. These files are compressed using the program AR (version 1.3). Any CoCo files (DECB) with the ".ARC" extension have been compressed with the TC program. In the future, a disk with AR 1.3 (OS-9 and OSK versions) and TC12 (for CoCo 1 and 2) and TC3 (for CoCo 3) on it will be sent with each subscription to "microdisk". Those ordering single issues will receive a utility disk with these programs on it FREE by request.

OS-9 Contents:

ar13 - ar version 1.3 compress/decompress utility (not compressed)
ar13.doc - ar 1.3 documentation file (not compressed)
sc.ar - contains sc (SuperComm version 2.2), *sc1con.icn* (Multi-View Icon for sc), and *scAIF.txt* (Multi-View AIF file)
sc22docs.ar - documentation for SuperComm version 2.2
bitbang.ar - complete bitbang driver, patches, and source code.
os9osk.ar - contains get_mouse, cocomouse_key, and OSK_signals text listings from "OS-9/OSK Answers!"

Color Computer Contents:

FUELCONS.BAS (Beginner's Show-PRINTLCT.BAS case Programs)
TC12, TC3 - archiver for CoCo 1/2, 3
VTO.BAS - Video Tape Organizer
PUPPO.ROM - ROM code for Puppo keyboard adapter
GETERM - Greg-E-Term terminal program. Up to 2400 baud, works with 32 through 80 column screens.

< 268'm >

< 268'm >

< 268'm >

New Lower Prices! from ColorSystems

Variations of Solitaire

Includes FIVE variations; Pyramid, Klondike, Spider, Poker and Canfield. Complete documentation shows how to create your own games boot disk using the special menu program which is included.

CoCo 3 Version \$29.95
MM/1 Version \$39.95

WPShel

A Word Processing Oriented Point and Click Shell for all your word processing needs. Requires WindInt from your Multi-Vue disk. Does not include Editor, Formatter, or Spelling Checker.

CoCo 3 Only! \$20.00

We accept Personal Checks or Money Orders drawn from US Banks or International Postal Money Orders. NC residents please add 6% Sales Tax. Call or write for FREE catalog! Please add \$3 per item for shipping outside of the Continental United States.

Quality OS-9 Software for the Color Computer 3 and the MM/1 from IMS

NEW!

Using AWK With OS-9

A description of the AWK Programming Language with an emphasis on GNU AWK for OSK. Includes the latest version of GNU AWK.

OSK ONLY! Just \$19.95

OS-9 Game Pack

Includes FIVE complete games; Othello, Yahtzee, Minefield, Knights Bridge, and Battleship. Includes special menu and step by step instructions on creating your own games boot disk.

CoCo 3 Version \$29.95
MM/1 Version \$39.95

All CoCo 3 Programs require at least 256K of memory.

Coming SOON! Indexed Files for OS-9 Level 2, OS-9/68000, and OS-9000!

ColorSystems

P.O. Box 540
Castle Hayne, NC 28429
(916) 675-1706

Paint version 1.0!

(Hyper-Tech Software) - \$54.00

This first commercial release of the classic PD program included with all MM/1 systems now offers file I/O for GIF and BRUN files. Edits images of 320 x 200 pictures. BRUN files are compiled into FLI animations using tools from the PIXUtils package. This is introductory priced, so buy NOW!

PIXUtils

(Hyper-tech Software) - \$19.00

Includes new and updated MM/1 utils for handling graphics images! AAPlay can now take FLI files apart and SnapIT saves a screen to disk in 2 formats. Also included are updated versions of GIFShow, GIFtoIFF, IFFShow, and the new CIFF2GIF utility. Once again, buy now for low, low introductory prices!

DESKTOP!

(Hyper-Tech Software) - \$54.00

A full featured ICON based GUI for the MM/1! Including it's own icon editor, and taking maximum advantage of the MM/1 windowing system this is one sweet system, and the bargain pricing WON'T LAST!

BlackHawk Enterprises

P.O. Box 10552, Enid, OK 73706-0552

See us for all your OS-9 6809 & 68000 needs!

Authorized MM/1 Distributor

AT&T (405) 234-2347 - Internet nimitz@delphi.com

Bob van der Poel Software

Great stuff for your OS-9 Level II system!

Ved Text Editor	\$24.95
Vprint Text Formatter	\$29.95
OS-9 character Set Editor	\$19.95
OS-9 Disk Mailing List (DML9)	\$24.95
Basic09 Subroutine Package	\$24.95
Cribbage	\$19.95
Ultra Label Maker	\$19.95
Magazine Index System	\$19.95
RMA Assembler Library	\$19.95
Stock Manager	\$24.95
OS-9 Public Domain Disk	\$9.95

(see DML9 review in this issue!)

All our programs are in stock for immediate shipping. Please include check or money order with your order. Sorry, no credit cards; but will ship COD to US and Canada (we add a small additional charge to cover the post office COD fee). Mention this ad and get FREE SHIPPING (normally 5% or \$2 minimum)! All orders are shipped via first class mail, usually the day received. Write or call for free DECB or OS-9/6800 catalogue.

P.O. Box 355
Porthill, ID
US 83853

P.O. Box 57
Wynndel, BC
Canada V0B 2N0

Telephone (604) 866-5772

for all your CoCo hardware needs, connect with

CoNect

449 South 90th Street
Milwaukee, WI 53214
414-258-2989 (after 5pm EST)

Mini RS-232 Port: Don't let the name fool you! This is a full featured serial port, supporting the signals needed for flow control as well as the basic 4. Jumper blocks allow readdressing or swapping DSR/DCD. No custom cables or hardware widgets needed here! Y cable users will need to add \$9.95 for a power supply. **\$49.95**

XPander: Don't you think the CoCo would be a lot nicer without all that mess hanging off the right side? Of course it would! Our XPander allows mounting two SCS decoded devices (like a floppy and hard drive controller) inside your CoCo. Built-in no-slot RS-232 port is similar to our "Mini" described above. The external cartridge connector is still present, and can be configured to run games or as an additional hardware slot. Kit includes new lower case shell and 12V power supply. Board only is great for use in a PC case!

Kit: \$124.95 Board Only: \$99.95

Hitachification: CoNect will install a Hitachi 63B09E CPU and a socket into your CoCo. Machine MUST be in working condition! The 68B09E will be returned unharmed. 90 day limited warranty. Chip and installation only **\$29.95**

We also have cables! Ask for a CoNect catalog.

Your Products Should be Here!

Advertising budget tight? Don't think you can afford to advertise? Well, you can't sell if you don't!
You can advertise in "the world of 68' micros"
for as little as \$10 per issue!

That's how much a 1/6 page (2"x 4") ad costs!

A 1/4 page (3 1/4"x 4 1/2") ad is only \$16.

See inside front cover for regular ad rates.

NOTE: These rates for first time advertisers only. Limited to three insertions.

Submitting Material to 68' micros

FARNA Systems retains rights to print and distribute any and all contributions. The submitter retains rights to distribute, but not to print in another publication without consent of FARNA Systems..

We accept program submissions in any programming language for DECB and OS-9 (6809 & 68000) of any type (games, utilities, etc.). Articles are accepted covering any aspect of Motorola 68xx and 68xxx processors. This includes microcontroller projects as well as alternate operating systems. If there are enough subscribers interested, we will begin accepting programs for alternate operating systems as well.

Submissions should be sent on disk in ASCII and executable formats. A letter describing the program or article should also be included. Submissions can be made to DSRTFOX on Delphi, or dsrtfox@delphi.com via Internet.

Media accepted: 5.25" disk in CoCo OS-9 (35/40T, SS/DS), IBM (DD/HD), or DECB (35/40 T). 3.5" in IBM only (DD/HD)

Have you ever wondered how nice it would be if you could lift an objection your screen out of it's surroundings by just pointing at it and clicking your mouse button? Or resize the working area on your screen that way? Now you can do that and a lot more with Level II Graphics.

But things don't stop there. *You can save your objects in disk files, print them as clipart or paste them into another screen*, even if this screen is a *different type* then the original. *Clipboard* functions also include stamp and reshape functions. You can also have multiple boards open simultaneously as long as you can find the memory for them.

Another part of Level II Graphics is the *Canvas* program. Here you can make your own artworks with all the tools you would expect: brushes, spraycans, drawtools, etc. The program also includes zoom, palette and animation functions. On top of that you have the entire screen as working area. File format is standard VEF and includes data compression.

Other programs included in the package are a *slideshow* program and *printer drivers for black/white printouts and for color printouts* that should work with most printers.

You will also find a *font editor* and a *pattern editor* to make your own fill patterns.

There is still more: *programs showing fractals from the julia and mandelbrot sets*. A program that shows you how to simulate plant growth on your computer. *Graphics subroutines you can use* with your own programs and source code that you can use as an example. There is also an *installation routine to take care of the hardest parts of getting things to run reliably*.

All of this doesn't fit on a single disk so you will get 2, one SS and one DS. System requirements are 512K and OS9. A hires adapter and ramdisk are recommended. Includes 40 page manual.

All for \$ 34.95

To get a demo copy of Level II Graphics send \$5 (+\$3 S/H). *This amount will be credited when you purchase the complete package.*

Tandy's Little Wonder

the MOST COMPLETE reference ever written for the Color Computer!

Soft-bound book contains:

- 140 BIG 8 1/2"x 11" Pages
- History of the CoCo (25% of content)
- Club and BBS Listings
- Current Supporting Vendors
- Peripheral Details
- Operating System Descriptions
- Programming Languages
- Repairs
- Hardware Upgrades
- Schematics
- Historic Memoirs
- MUCH, MUCH MORE!

ONLY **\$25⁰⁰**

(add \$2.50 S&H, \$4 Canada, \$7 Overseas)

Say "I saw it in 68' micros" and get \$1 off!

FARNA Systems PB

Box 321, Warner Robins, GA 31099-0321

ADOS, ADOS 3, and Extended ADOS 3

the premier extensions to Disk Basic, are now available from FARNA Systems!

Original ADOS (CC 1 & 2) - \$15

ADOS 3 - \$25

Extended ADOS 3 - \$30 (ADOS 3 required)

ADOS-3/Extended ADOS 3 combo - \$50

add \$2.50 S&H, \$4 Canada, \$7 Overseas

Also available:

TOOLS 3 version 1.1 **\$29.95**

Quickletter version 2.0 **\$19.95**

Accounting level 2 **\$34.95**

Investing level 2 **\$24.95**

upgrades from earlier versions **\$5.00**

New products:

RScopy/OScopy - transfer files from OS-9 disks to DECB disks and vice-versa *without* loading special device drivers. **\$10**

Coming in October:

CoCoTop - a desktop program for OS9 level 2. 50+ file and system management functions in a point and click environment.

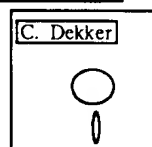
C. Dekker ... User-friendly Level II Programs!

RR #4 Centreville, NB

E0J 1H0, CANADA

Phone 506-276-4841

Shipping+handling: US/Canada \$3.00 all others \$5.00
Prices in US dollars Send cheque or money order NO COD'S



FARNA Systems

Box 321, Warner Robins, GA 31098-0321
(912)328-7859 - Internet dsrtfox@delphi.com
"The Color Computer Specialist"

CoCo Family Recorder 2.0 - \$20.00

The ULTIMATE genealogy program for the CoCo 3. Several reports, info gathering sheets, family tree, etc. Can print any screen. Up to 550 people, 300 marriages. Requires two drives and an 80 column monitor.

OS-9 Quick Reference Guide - \$7.50

Handy 5 1/2"x8 1/2", 42 page reference contains command syntax, special keys, editor commands, error codes, and system calls... everything the user or programmer needs in an easy desk size!

Patch OS-9 - \$7.50

This two disk set contains all the most recommended and helpful patches, commands, and utilities that make Tandy's OS-9 Level II really shine! A patch program installs all automatically. Two double sided drives (40 or 80 track) and 512K (minimum) required for auto patch utility. Instructions for manual patching on disk also.

Quick Reference Guide and Patch OS-9... both for \$12!

RASCAN/DigiScan Video Digitizer - \$150.00

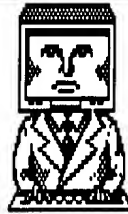
Convert still video images to computer images. Process takes about 15 seconds per screen. Plugs into joystick ports... no MPI required! Max-10, CoCo Max 3, and Color Max 3 compatible. Up to 640x200 resolution. Special \$20 handling fee applies. Shipped from Australia, allow 12 weeks delivery. 50% down, 50% before delivery. *Reviewed in Aug. '93*

Cross Road II - hi-res tic-tac-toe with sound! - \$10.00

Mind Games - collection of 9 classic games - \$10.00

Add \$2 per order for S&H (Canada \$4, others \$6)

COCO FRIENDS DISK MAGAZINE



COCO FRIENDS DISK MAGAZINE
(CFDM) is devoted exclusively to those who still enjoy running under "RS DOS". The standard system needed to use CFDM is a CoCo 3, RGB monitor, at least one disk drive, and the "RS Dos" which came with your CoCo 3

CFDM is a monthly disk based publication which is based on a "floppy" disk. When you "run" the "magazine" side of CFDM, you'll be greeted with a beautiful cover picture by CoCo Friend James Gibbons. Pressing any key takes you to the magazine's colorful Main Menu. There you'll find 14 sections which are filled with entries. Sections included are: About CFDM, About this Issue, Active CoCo, Advertisements, CoCo Friends Art Gallery, Articles of the Month, Family Tree, Forum, From the Editor, Letters to the Editor, Potpourri, Programs of the Month, Reviews, and Questions & Answers.

Next you will enter a section and find a number of entries written by our CoCo Friends from all over the world. Each issue of CFDM contains 60 to 80 entries. Some sections contain documentation about the many programs and graphics found on the "flip-side" of CFDM.

The "flip-side" or "program side" of CFDM is filled with contributions of wonderful programs and graphics from our many CoCo Friends! Each issue has from two to four hi-res pictures and from eight to fifteen never-before-seen programs.

The cost of CFDM is \$6 per issue or \$30 for a six issue subscription. Join us now for a lot of fun! If you're like me, there's never too many Friends around to enjoy a great time on the CoCo.

SEND CASH, CHECK, OR MONEY ORDER TO:

RICK'S COMPUTER ENTERPRISE

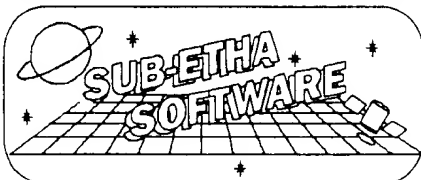
(606) 787-5783 CODs Add \$2.50
P.O. Box 276
Liberty, KY 42539

type writ er n. archaic machine for writing in characters by means of keyboard operated types striking through an inked ribbon.
right-write n. a full-featured word processor for the MM/1 computer. Has multiple fonts, WYSIWYG display of bold, italics, underline, and colors on screen, menu driven, cut and paste, on-screen ruler, much more! **\$59.95**

Mini Banners - Print single or multi line banners with ANY printer! Independantly sized text on each line. Menu driven, over 30 fonts. OS-9/DECB - \$19.95 MM/1 - \$24.95

CheckBook+ - Popular checkbook tracking/balancing program. Fully menu driven. Visualize your account through graphs! Pop-up calculatro with memory, cut and paste editing, more! Supports almost any printer. DECB requires mouse or joystick. OS-9/DECB - \$24.95 MM/1 - \$34.95

Write for a full product listing!



P.O. Box 152442, Lufkin, TX 75915

Please include \$2.50 S&H per order

ADVERTISER'S INDEX:

Atlanta CoCoFest	12
BlackHawk Enterprises	27
Bob van der Poel Software	27
C. Dekker	28
CoCo Friends Disk Magazine	29
Color Systems	27
CoNect	27
DISTO	25
FARNA Systems	28,29
Delmar Company	BC
Infinium Technology	20
Mid Iowa & Country CoCo	14
OS-9 Underground	9
Sub-Etha Software	29

New BBS!

"980 Main" 216-532-2742
Supports CoCo (DECB and OS-9)
CoCo Sysop is Jason Reighard
(writer of VTO in this issue)

He needs your support!

(Remember... it is cheaper to call long distance out of state!)

For superior OS-9 performance, the

SYSTEM V

Provides a 68020 running at 25 MHz, up to 128 MBytes of 0 wait-state memory, SCSI and IDE interfaces, 4 serial and 2 parallel ports, 5 16-bit and 2 8-bit ISA slots and much more. The SYSTEM V builds on the design concepts proven in the SYSTEM IV, providing maximum flexibility with inexpensive expandability.

AN OS-9 FIRST - the MICROPROCESSOR is mounted on a daughter board which plugs onto the motherboard. This will permit inexpensive upgrades in the future when even greater performance is required.

G-WINDOWS benchmark performance index is 0.15 seconds faster with a standard VGA board than a 68030 running at 30 MHz with ARTC video board (85.90 seconds vs 86.05 seconds).

Or, for less demanding requirements, the

SYSTEM IV

The perfect low cost, high-quality and high performance OS-9 computer serving customers world-wide. Designed for and accepted by industry. Ideal low-cost work-station, development platform, or just plain fun machine. Powerful, flexible, and inexpensively expandable. Uses a 68000 microprocessor running at 16 MHz.

Both computers provide flexible screen displays in the native mode with the optional VGA card.

Eight text modes are supported -

40 x 24	80 x 25
80 x 50	100 x 40
132 x 25	132 x 28
132 x 44	132 x 60

Foreground, background, and border colors are user selectable from up to 16 colors.

Eleven graphics modes are supported -

640 x 200 x 16	320 x 200 x 256
640 x 350 x 16	640 x 350 x 256
640 x 480 x 16	640 x 400 x 256
800 x 600 x 16	640 x 480 x 256
1024 x 768 x 16	800 x 600 x 256
	1024 x 768 x 256

Text and graphics modes may be selected by a utility provided, MODESET, by software using SetStt calls, or by termcap entries. In the text mode, the screen responds to standard VT100 control sequences. The full character set from Hex 20 through Hex FF is supported in the text modes up to and including 100 characters wide. The upper 128 characters follow the 'IBM Character Set 2' popular with many terminals and printers. These may be displayed on the screen by using the 'Alt' key and one or two other keys (software permitting).

G-WINDOWS option provides three screen resolutions: 640 x 480 x 256, 800 x 600 x 256, or 1024 x 768 x 256. You can have two full size 80 x 25 windows with room to spare. A window as large as 122 x 44 using the large fonts or over 180 x 70 using the small fonts is also possible.

delmar co

PO Box 78 - 5238 Summit Bridge Road - Middletown, DE 19709
302-378-2555 FAX 302-378-2556